



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

**Grau en Enginyeria Electrònica Industrial i Automàtica**

**POSADA EN MARXA D'UN ROBOT COL·LABORATIU**



**Memòria i Annexos**

<b>Autor:</b>	Bisnu Masó Clota
<b>Director:</b>	Pedro Ponsa Asensio
<b>Co-Director:</b>	Sebastián Tornil Sin
<b>Convocatòria:</b>	Juny 2020



## Resum

Aquest document és el resultat del treball de final d'estudis del grau d'Enginyeria Electrònica Industrial i Automàtica. Té la finalitat de realitzar la posada en marxa d'una estació amb robot col·laboratiu de la marca Universal Robots i el model UR3e per aplicacions docents.

Seguint la metodologia exposada a la introducció es dur a terme un estudi teòric sobre els robots industrials i les normatives i legislacions que regulen la seguretat d'aquests robots. Tot seguit s'introdueix en la robòtica col·laborativa fent èmfasis al robot UR3e del qual s'estudien les característiques, els tipus de comunicacions i la programació. Seguidament, passant a nivell més pràctic, es detallen els processos de la instal·lació del robot, el disseny i construcció de complements i la seva posada en marxa. Es fa un estudi del sensor de força i com utilitzar-lo i, a continuació, es posen en pràctica, amb el simulador URsim, aplicacions amb comunicació utilitzant Matlab i el protocol Ethernet Sockets del robot. Finalment, es fa un estudi sobre la col·laboració humà-robot que pretén desenvolupar elements que facilitin aquesta interacció, es creen diferents manuals d'ajuda per la docència i es finalitza amb un cas pràctic que es va celebrar en les jornades de portes obertes del centre on es mostra un programa i el seu funcionament.

## Resumen

Este documento es el resultado del trabajo de final de estudios del grado de Ingeniería Electrónica Industrial y Automática. Tiene la finalidad de llevar a cabo la puesta en marcha de una estación con robot colaborativo de la marca Universal Robots y el modelo UR3e para aplicaciones docentes.

Siguiendo la metodología expuesta en la introducción se llevó a cabo un estudio teórico sobre los robots industriales y las normativas y legislaciones que regulan la seguridad de estos robots. A continuación, se introduce en la robótica colaborativa haciendo énfasis al robot UR3e del que se estudian las características, los tipos de comunicaciones y la programación. Seguidamente, pasando a nivel más práctico, se detallan los procesos de la instalación del robot, el diseño y construcción de complementos y su puesta en marcha. Se hace un estudio del sensor de fuerza y cómo utilizarlo y, a continuación, se ponen en práctica, con el simulador URsim, aplicaciones con comunicación utilizando Matlab y el protocolo Ethernet Sockets del robot. Finalmente, se hace un estudio sobre la colaboración humano-robot que pretende desarrollar elementos que faciliten esta interacción, se crean diferentes manuales de ayuda para la docencia y se finaliza con un caso práctico que se llevó a cabo en las jornadas de puertas abiertas del centro donde se muestra un programa y su funcionamiento.

## **Abstract**

This thesis is the result of the final degree project of the Bachelor's degree in Industrial Electronics and Automatic Control Engineering. Its purpose is to develop the implementation of a robotic cell for teaching applications based on an UR3 (Universal Robots) collaborative robot.

Following the methodology outlined in the introduction, a theoretical study on industrial robots and the regulations and legislation governing the safety of these robots is realized. Afterwards, collaborative robots are introduced, making emphasis on the UR3e robot, studying their characteristics, types of communications and programming languages. Then, moving into a more practical level, the processes of robot installation, the design and construction of accessories and the set up process are detailed. Following that, a study of the force sensor and how to use it is made, applications about communication are put into practice with URsim simulator using Matlab and the robot's Ethernet Sockets protocol. Finally, there is a study about human-robot collaboration that aims to develop elements that facilitate this interaction, different teaching aid manuals are created and it ends with a practical case study that was carried out in the centre's open days where a programme and its operation is shown.

## Agraïments

En primer lloc agrair al director i co-director del TFG, Pere Ponsa i Sebastián Tornil, per donar-me l'oportunitat de poder treballar amb aquest robot. També agrair-los per la seva disponibilitat, l'atenció i la dedicació constant per resoldre dubtes i per guiar-me en tot moment al llarg de l'elaboració d'aquest treball.

Per altra banda, també agrair als mestres de taller del departament d'electrònica per facilitar-me tot el material i per oferir-se a ajudar-me en el procés de muntatge de l'estació del robot.

Finalment, agrair a tots els amics i familiars que heu aportat el vostre granet de sorra pel desenvolupament d'aquest treball de final de grau.

## Glossari

### **AENOR**

*Asociación Española de Normalización y Certificación*

### **CAS**

*Collaborative Assembly Systems*

### **CIP**

*Common Industrial Protocol*

### **DHCP**

*Dynamic Host Configuration Protocol*

### **FAS**

*Flexible Assembly Systems*

### **FTP**

*File Transfer Protocol*

### **GdL**

*Graus de Llibertat*

### **GND**

*Ground (terme elèctric per referir-se al terra)*

### **GUI**

*Grafic User Interface*

### **HDL**

*Hardware Description Language*

### **HRC**

*Human Robot Colabotarion*

### **ISO**

*International Organization for Standardization (Organització Internacional d'Estandardització)*

### **MAS**

*Manual Assembly Systems*

### **NAT**

*Network Address Translation*

### **PL**

*Performance Level*

### **PolyScope**

*Consola de programació, Teach pendant*

### **RIVC**

*Robòtica Industrial i Visió per Computador*

### **SCARA**

*Selective Compliance Assembly Robot Arm*

### **SIL**

*Safety Integrity Level*

### **TCP**

*Terminal Center Point*

### **TÜV**

*Technischer Überwachungs Verein ( Associació d'Inspecció Tècnica)*

### **UNE**

*Una Norma Española*



**VAC**

*Voltatge Corrent Altern*

**VDC**

*Voltatge Corrent Continu*





# Índex

<b>RESUM</b>	<b>I</b>
<b>RESUMEN</b>	<b>II</b>
<b>ABSTRACT</b>	<b>III</b>
<b>AGRAÏMENTS</b>	<b>IV</b>
<b>GLOSSARI</b>	<b>V</b>
<b>ÍNDEX D'IL·LUSTRACIONS</b>	<b>11</b>
<b>ÍNDEX DE TAULES</b>	<b>13</b>
<b>1. INTRODUCCIÓ</b>	<b>14</b>
1.1. Objectius del treball.....	14
1.2. Abast del treball.....	15
<b>2. ROBÒTICA</b>	<b>17</b>
2.1. Definició .....	17
2.2. Classificació .....	17
2.3. Robots industrials .....	18
2.3.1. Robots col·laboratius .....	19
2.4. Característiques bàsiques.....	22
2.5. Legislació.....	24
2.6. Seguretat.....	25
2.6.1. Seguretat robots col·laboratius .....	27
<b>3. ELECCIÓ DEL ROBOT</b>	<b>28</b>
3.1. Estat de l'art dels robots col·laboratius.....	28
3.2. Universal Robots.....	28
3.2.1. UR3e - Especificacions .....	29
3.2.2. Ports d'E/S.....	32
3.2.3. Comunicacions .....	33
3.2.4. Llenguatge de programació .....	37
3.2.5. Entorn de programació extern i simulació: .....	40
<b>4. MUNTATGE ROBOT</b>	<b>43</b>
4.1. Disposició ( <i>layout</i> ) .....	43

4.2.	Instal·lació .....	44
<b>5.</b>	<b>DISSENY I CONSTRUCCIÓ DE COMPLEMENTS .....</b>	<b>47</b>
5.1.	Element terminal .....	47
5.1.1.	Versió 1 .....	47
5.1.2.	Versió 2 .....	49
5.2.	Simulador sensors digitals .....	50
5.3.	Muntatge sistema elèctric de connexions.....	51
5.4.	Construcció suport consola de programació.....	52
<b>6.</b>	<b>POSADA EN MARXA .....</b>	<b>53</b>
6.1.	Seguretat .....	53
6.2.	Determinació del nou TCP .....	55
<b>7.</b>	<b>ESTUDI DE LA FORÇA .....</b>	<b>57</b>
7.1.	Transductors de força .....	57
7.2.	UR3e i la força .....	57
7.3.	Aplicacions docents.....	61
<b>8.</b>	<b>ESTUDI DE LA COMUNICACIÓ I ACCÉS REMOT .....</b>	<b>62</b>
8.1.	Accés remot fixers .....	62
8.2.	Accés remot control robot.....	64
8.2.1.	Comandes bàsiques: Dashboard Server .....	64
8.2.2.	Comunicació avançada: Sockets .....	64
<b>9.</b>	<b>APLICACIÓ AMB VISIÓ .....</b>	<b>67</b>
9.1.	Funcionament .....	68
9.2.	Codi.....	69
9.2.1.	Programa Matlab.....	69
9.2.2.	Programa robot .....	71
<b>10.</b>	<b>COL·LABORACIÓ HUMÀ-ROBOT .....</b>	<b>75</b>
10.1.	Aplicacions col·laboratives.....	77
<b>11.</b>	<b>CAS PRÀCTIC: JORNADES PORTES OBERTES .....</b>	<b>79</b>
11.1.	Disseny i distribució .....	79
11.2.	Funcionament .....	81
11.3.	Programa.....	83
<b>12.</b>	<b>MATERIAL D'AJUDA A LA DOCÈNCIA .....</b>	<b>85</b>

12.1. Instal·lació màquina virtual i URsim .....	85
12.2. Creació de programes.....	86
12.3. Configuració comunicació URsim (VM).....	90
12.4. Transferència programes entre USB-URsim i viceversa. ....	94
<b>CONCLUSIONS</b> .....	<b>95</b>
<b>PRESSUPOST I/O ANÀLISI ECONÒMICA</b> .....	<b>97</b>
<b>BIBLIOGRAFIA</b> .....	<b>101</b>
<b>ANNEX</b> .....	<b>105</b>
A1. Codi Matlab complet capítol 9.2.1.....	105
A2. Plànols .....	108
A3. Esquemes elèctrics.....	108
A3. Datasheet Robot Ur3e .....	108

## Índex d'il·lustracions

<b>Figura 2.1.</b> Classificació de la col·laboració entre robot-operari (Font: [9]) .....	19
<b>Figura 2.2.</b> Representació de parada monitoritzada (Font: [10]).....	20
<b>Figura 2.3.</b> Representació guiata manual (Font: [10]).....	20
<b>Figura 2.4.</b> Representació monitorització de velocitat i separació (Font: [10]).....	21
<b>Figura 2.5.</b> Representació limitació de potència i força (Font: [10]).....	21
<b>Figura 2.6.</b> Representació cadena cinemàtica oberta i tancada (Font: Bisnu Masó).....	22
<b>Figura 2.7.</b> Representació ares de treball d'un robot cartesià, scara i articulata (Font: [12]).....	23
<b>Figura 2.8.</b> Tipus de repetibilitat i precisió (Font: Bisnu Masó).....	23
<b>Figura 2.9.</b> Diferència entre les normatives i les directives en format gràfica (Font: Bisnu Masó).....	24
<b>Figura 2.10.</b> Nous riscos que comporten els robots col·laboratius (Font: [21]).....	27
<b>Figura 3.1.</b> Determinació graus de llibertat robot UR3e (Font: Bisnu Masó) .....	29
<b>Figura 3.2.</b> Àrea de treball robot UR3e (Font: [1]) .....	30
<b>Figura 3.3.</b> Gràfica capacitat de càrrega respecte el centre de gravetat (Font: [1]) .....	31
<b>Figura 3.4.</b> Interior controlador robot UR3e i els pots de connexió (Font: [2]).....	32
<b>Figura 3.5.</b> Estació bàsica amb el braç robòtic, la consola de programació i la caixa del controlador (Font: Bisnu Masó).....	33
<b>Figura 3.6.</b> Plataforma MIR100 amb robot UR5e (Font: [32]).....	36
<b>Figura 3.7.</b> Diagrama de flux de la "missió" del robot MIR (Font:[32]) .....	37
<b>Figura 3.8.</b> A l'esquerra programació Scratch, a la dreta programació intuïtiva UR (Font: Bisnu Masó).....	38
<b>Figura 3.9.</b> Captura pantalla de programació UR (Font: Bisnu Masó) .....	38
<b>Figura 3.10.</b> Fitxers generats amb la programació intuïtiva (Font: Bisnu Masó).....	39
<b>Figura 3.11.</b> Exemple seqüència programació intuïtiva (Font: Bisnu Masó) .....	39
<b>Figura 3.12.</b> Captura màquina virtual i simulador URSim (Font: Bisnu Masó).....	41
<b>Figura 3.13.</b> Captura pantalla I/O i mode simulació activat (Font: Bisnu Masó) .....	42
<b>Figura 4.1.</b> Distribució i Layout aula A5.4 amb la nova estació UR3e (Font: Bisnu Masó).....	44
<b>Figura 4.2.</b> Distribució robot en la taula de treball (Font: Bisnu Masó).....	45
<b>Figura 4.3.</b> Col·locació i muntatge platina suport robot (Font: Bisnu Masó).....	45
<b>Figura 4.4.</b> Muntatge i fixació robot amb la platina (Font: Bisnu Masó) .....	46
<b>Figura 4.5.</b> Caixa controlador robot UR3e (Font: Bisnu Masó).....	46
<b>Figura 5.1.</b> Disseny element terminal per electroimant (Font: Bisnu Masó).....	47
<b>Figura 5.2.</b> Instal·lació element terminal i visió final (Font: Bisnu Masó).....	48
<b>Figura 5.3.</b> Esquema elèctric connexió electroimant (Font: Bisnu Masó).....	48
<b>Figura 5.4.</b> Peça de fusta amb planxa metàl·lica (Font: Bisnu Masó) .....	49
<b>Figura 5.5.</b> Versió final de l'element terminal amb adaptador d'ús genèric. (Font: Bisnu Masó).....	50
<b>Figura 5.6.</b> Element terminal amb indicador lluminós en funcionament (Font: Bisnu Masó).....	50
<b>Figura 5.7.</b> Versió inicial simulador sensors digitals (Font: Bisnu Masó).....	51
<b>Figura 5.8.</b> Caixa de connexions banana d'I/O del controlador (Font: Bisnu Masó).....	52
<b>Figura 5.9.</b> Disseny i construcció suport consola de programació (Font: Bisnu Masó) .....	52

<b>Figura 6.1.</b> Captura definició pla de seguretat vertical (Font: Bisnu Masó).....	53
<b>Figura 6.2.</b> Captura com afegir nous plans (Font: Bisnu Masó).....	54
<b>Figura 6.3.</b> Plans de seguretats definits en el robot UR3e (Font: Bisnu Masó) .....	54
<b>Figura 6.4.</b> Canvi icona suma de comprovació al afegir plans de seguretat (Font: Bisnu Masó) .....	55
<b>Figura 6.5.</b> Diferència de recorregut amb i sense TCP definit (Font: [34]) .....	55
<b>Figura 6.6.</b> Captura definició TCP i altes paràmetres de l'element terminal (Font: Bisnu Masó) .....	56
<b>Figura 7.1.</b> Eixos de detecció força i parell del sensor de força (Font: Bisnu Masó).....	58
<b>Figura 7.2.</b> Captura funció Screwdriwing i paràmetres a ajusta (Font: Bisnu Masó).....	59
<b>Figura 7.3.</b> Eixos i sentit de la força del mode simple (Font: Bisnu Masó) .....	59
<b>Figura 7.4.</b> Exemple d'us del mode de força frame (Font: Bisnu Masó).....	60
<b>Figura 7.5.</b> Exemple d'us del mode de força point (Font: Bisnu Masó).....	60
<b>Figura 7.6.</b> Línies de codi per la lectura de la força amb programació intuïtiva (Font: Bisnu Masó).....	61
<b>Figura 8.1.</b> Captura vídeo demostratiu del funcionament de la comunicació (Font: Bisnu Masó).....	66
<b>Figura 9.1.</b> Disseny peces de diferents formes per la visió per computador (Font: Bisnu Masó).....	67
<b>Figura 9.2.</b> Sistema de captura i transport de peces per l'aplicació de visió(Font: Bisnu Masó) .....	68
<b>Figura 9.3.</b> Estació completa per l'aplicació de visió per computador (Font: Bisnu Masó).....	68
<b>Figura 9.4.</b> Diagrama funcionament aplicació de visió per computador (Font: Bisnu Masó) .....	69
<b>Figura 9.5.</b> Representació gràfica dels punts de pas i punts d'actuació aplicació de visió (Font: Bisnu Masó).....	74
<b>Figura 10.1.</b> Disseny i distribució elements panell de comandament (Font: Bisnu Masó).....	75
<b>Figura 10.2.</b> Disseny i distribució panell auxiliar (Font: Bisnu Masó) .....	76
<b>Figura 10.3.</b> Disseny i distribució elements panell alternatiu (Font: Bisnu Masó) .....	77
<b>Figura 10.4.</b> Distribució sistema flexible d'assemblatge(FAS) (Font:[40]).....	78
<b>Figura 11.1.</b> Distribució àrea de treball amb les tres zones (Font: Bisnu Masó) .....	79
<b>Figura 11.2.</b> Primeres proves de disseny d'estació i prototips (Font: Bisnu Masó) .....	80
<b>Figura 11.3.</b> Disseny final d'estació amb les tres tasques diferents (Font: Bisnu Masó).....	81
<b>Figura 11.4.</b> Botonera selecció de tasca i zona de treball (Font: Bisnu Masó).....	81
<b>Figura 11.5.</b> Esquema elèctric i ports de connexió botonera (Font: Bisnu Masó).....	82
<b>Figura 11.6.</b> Grafset de nivell 1 del funcionament del programa (Font: Bisnu Masó) .....	83
<b>Figura 12.1.</b> Icona VMware (Font: Bisnu Masó).....	85
<b>Figura 12.2.</b> Menú selecció per descarregar URsim (Font: Bisnu Masó).....	85
<b>Figura 12.3.</b> Descompressió carpeta descarregada (Font: Bisnu Masó).....	86
<b>Figura 12.4.</b> Captura instal·lació URsim a VMware (Font: Bisnu Masó).....	86
<b>Figura 12.5.</b> Captura pantalla de programació UR (Font: Bisnu Masó).....	87
<b>Figura 12.6.</b> Programa d'exemple d'un programa senzill. (Font: Bisnu Masó).....	89
<b>Figura 12.7.</b> Mètodes per posicionar el robo (Font: Bisnu Masó) .....	90
<b>Figura 12.8.</b> Captura configuració xarxa i IP per la comunicació (Font: Bisnu Masó).....	91
<b>Figura 12.9.</b> Captura adaptadors de xarxa ordinador (Font: Bisnu Masó) .....	91
<b>Figura 12.10.</b> Captura paràmetres de configuració adaptador de xarxa ordinador (Font: Bisnu Masó).....	92
<b>Figura 12.11.</b> Captura accés configuració d'adaptador de xarxa màquina virtual(Font: Bisnu Masó).....	92
<b>Figura 12.12.</b> Captura configuració adaptador de xarxa màquina virtual (Font: Bisnu Masó).....	93
<b>Figura 12.13.</b> Captura habilitació mode remot robot físic (Font: Bisnu Masó).....	93

## Índex de taules

<b>Taula 2.1.</b> Paràmetres cadena cinemàtica. ....	22
<b>Taula 2.2.</b> Nivell de fiabilitat i la seva probabilitat [18]. ....	26
<b>Taula 2.3.</b> Nivell d'integritat de seguretat [19]. ....	26
<b>Taula 3.1.</b> Comparativa robots col·laboratius. ....	28
<b>Taula 3.2.</b> Resum característiques UR3e. ....	31
<b>Taula 5.1.</b> Especificacions electroimant XRN-XP. ....	48
<b>Taula 10.1.</b> Descripció color indicador lluminós amb relació a l'estat de l'estació. ....	76
<b>Taula 12.1.</b> Descripció de comandes per la programació intuïtiva. ....	88
<b>Taula 0.1.</b> Llista materials i els costos associats. ....	97
<b>Taula 0.2.</b> Costos amortitzables. ....	99
<b>Taula 0.3.</b> Costos recursos humans. ....	99
<b>Taula 0.4.</b> Costos totals del projecte. ....	100

# 1. Introducció

L'adquisició d'un robot col·laboratiu de la marca Univerasal Robot pel departament d'ESAI (Enginyeria de Sistemes, Automàtica i Informàtica Industrial) suposa la primera incorporació d'un robot d'aquest tipus i marca en el centre docent. A conseqüència, apareix la necessitat d'investigar el funcionament i elabora contingut que pugui servir d'utilitat per a la docència.

El motiu per l'elecció d'aquest treball fou l'oportunitat presentada, pel director del treball, de poder treballar, des de zero, amb un robot actual i innovador, fet que permet desenvolupar els meus coneixements i formar-me per oportunitats futures donat que aquests tipus de robots tenen una acceptació cada vegada major en la indústria i els petits negocis.

La metodologia emprada pel desenvolupament del treball es pot dividir en quatre fases: la primera fase correspon en la recerca d'informació i funcionament del robot en l'àmbit teòric, la segona fase consisteix en el muntatge del robot, la tercera fase en la programació i simulació dels codis i, la quarta i última fase, es dur a terme l'elaboració de manuals i documentació per ús acadèmic. Els recursos utilitzats pel desenvolupament d'aquest treball, entre d'altres, són: UR3e manual d'usuari [1], UR Academy [2], Zacobria (web de suport a la comunitat UR) [3], IEEE Xplore [4], etc.

## 1.1. Objectius del treball

L'objectiu principal d'aquest treball és la posada en marxa d'una estació amb un robot col·laboratiu UR3e per aplicacions docents. Per assolir aquest objectiu s'han detallat objectius més específics que es presenten a continuació i s'aborden en els següents capítols:

- Muntatge del robot en el laboratori de robòtica i automàtica A5.4 minimitzant el moviment dels equips actuals i maximitzant l'espai lliure disponible.
- Disseny i construcció d'accessoris que permetin tenir un robot funciona i una estació completa.
- Posada en marxa del robot i configuració dels paràmetres de seguretat.
- Recerca sobre el sensor de força i les seves utilitats i funcionalitats.
- Estudi de les comunicacions i programació d'aplicacions demostratives.
- Estudi de tasques col·laboratives humà-robot.
- Creació de material docent en forma de manuals i programes d'exemple una vegada après el funcionament del robot.



## **1.2. Abast del treball**

A partir de la metodologia citada a la introducció, en finalitzar el treball, es pretén aconseguir disposar d'una estació amb el robot UR3e funcional i documentació bàsica per a la seva programació i la iniciació en aquest tipus de robots i, concretament, del model que es disposa. Aquest material està dirigit per la docència i pels futurs estudiants que tindran l'oportunitat de poder-se formar amb la robòtica col·laborativa.

A causa de les circumstàncies excepcionals causades per la COVID-19, tots els programes i aplicacions s'ha provat amb el simulador URsim i no s'han pogut testear amb el robot físic. Per tant, una de les futures tasques podria ser comprovar el correcte funcionament dels programes creats amb el robot UR3e del laboratori.

La programació i desenvolupament de programes més avançats, com la recerca d'aplicacions més complexes, poden ser futures propostes de treballs de final d'estudis, tant de grau com de màsters.



## 2. Robòtica

### 2.1. Definició

La robòtica és la disciplina encarregada de l'estudi, el disseny i la construcció de robots. Un robot, parlant amb termes generals, és una màquina o dispositiu programable capaç, de forma automàtica, dur a terme una o varies tasques les quals, anteriorment, les efectuaven les persones.

L'origen d'aquest mot prové de la llengua esclava txeca on s'utilitzava la paraula *robota* per designar la servitud, esclavitud o el treball forçat a mitjans del segle XIX. No va ser fins al següent segle, en l'obra de teatre *Robots Universals de Rossum*, Karel Čapek, quan es va utilitzar el terme robot amb el significat que coneixem avui dia [5].

### 2.2. Classificació

Segons els camps d'aplicació de la robòtica es poden classificar en dos grans grups:

- **Robots industrials:** varen sorgir per la necessitat d'augmentar la producció industrial i disminuir els costos de fabricació. Comunament s'utilitzen per realitzar tasques molt repetitives, tasques que un humà no pot dur a terme o tasques que poden suposar un risc per les persones.
- **Robots de servei:** tots aquells robots que s'utilitzen fora dels àmbits industrials i, normalment, faciliten serveis als humans, des de robots d'investigació i recerca, assistència de persones i entreteniment d'aquestes [6]. Els més comuns d'aquest grup són [7]:
  - **Humanoides:** robots amb aspecte humà que tendeixen a copiar el comportament de l'ésser humà.
  - **Cotxes autònoms:** molt popularitzat en aquests últims anys, són dispositius capaços de circular per la carretera sense la intervenció humana amb una lectura i presa de decisions immediata.
  - **Drons i submarins:** vehicles no tripulats de dimensions diverses, en el primer cas són dispositius aeris i en el segon dispositius aquàtics molt utilitzats per la recerca i investigació.
  - **Exoesquelets:** tal com indica el nom, és un dispositiu que s'adapta al cos humà i permet reduir els esforços en realitzar un moviment o un força física. S'utilitza per a la rehabilitació de persones i, actualment, també es comença emprar en àmbits industrials.

- **Robots de consum:** són dispositius destinats a l'entreteniment, sovint joguines per als nens i nenes, o per tasques domèstiques com netejar o fregar. No són programables.
- **Robots educatius:** dispositius programables que s'utilitzen a les aules dels centres educatius o a casa i fomenten l'aprenentatge de l'usuari.
- **Robots mèdics:** dispositius utilitzats en l'àmbit sanitari com els robots quirúrgics o els supercomputadors utilitzats per simular al tenir una capacitat de càlcul i resolució d'algoritmes molt elevada.
- **Robots zoomòrfics:** dispositius que, igual que els humanoides, tenen aspecte d'animals i reproduïxen el seu comportament.
- **Robots militars,** etc.

## 2.3. Robots industrials

Tal com s'ha explicat en l'apartat anterior, un robot industrial és una màquina programable capaç de fer qualsevol tasca per la qual ha estat dissenyada i programada. Segons l'estructura i la tipologia del robot es poden diferenciar entre [8]:

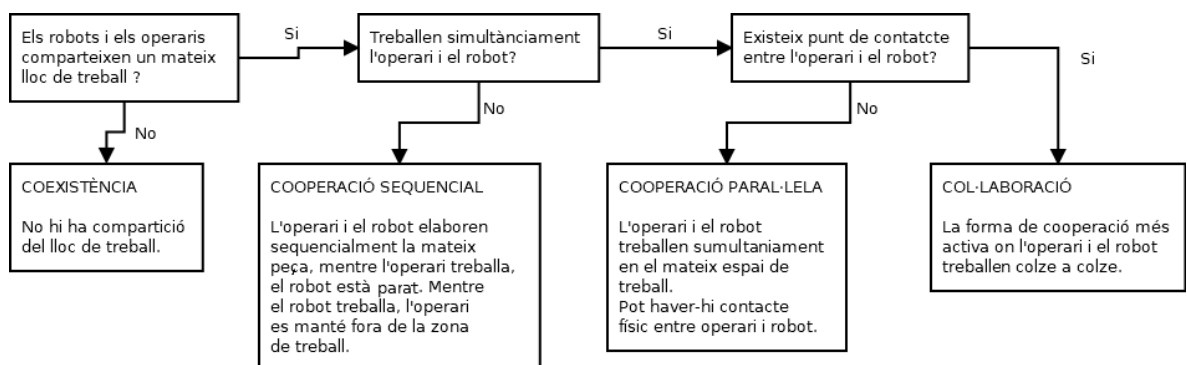
- **Robots cartesianes:** són robots lineals constituïts per tres eixos i que els permet moure pel sistema de coordenades cartesianes com el seu nom indica (X, Y, Z). Gràcies a la tipologia és un dispositiu flexible quant a velocitat, precisió i mida del robot. És un sistema comú en les màquines de control numèric per ordinador (CNC)
- **Robots SCARA:** provinent de l'acrònim en anglès *Selective Compliance Assembly Robot Arm* són robots amb l'espai de treball igual que els cartesianes però totes les articulacions són de revolució i una articulació prismàtica i destaquen per la rapidesa dels moviments i la precisió. Són molt utilitzats en aplicacions de paletització i desplaçaments d'objectes.
- **Robots cilíndrics:** molt semblants als SCARA però consten només d'un eix de revolució i dos eixos prismàtics. Solen ser de dimensions compactes i utilitzats en aplicacions senzilles d'assemblatge de peces.
- **Robots delta:** també coneguts com a robots paral·lels, són robots formats per tres braços connectats en un punt comú. La posició de la base de l'element terminal ve determinat pels moviments simultanis dels tres braços. Degut a les limitacions, l'àrea de treball és menor però permet moure's a altes velocitats i precisió. Utilitzat en indústries alimentàries per la selecció i col·locació dels productes.
- **Robots polars:** són robots amb 3 moviments, dues articulacions de rotació en la base i un braç amb una articulació lineal. Com indica el nom, treballen a partir de coordenades polars. Van ser un dels primers models de robots utilitzats per la manipulació de materials, a la indústria de la fosa i treball de metalls, etc.

- **Robots articulats:** són robots de tres o més eixos que permeten efectuar moviments més complexos que s'aproximen als moviments d'un braç humà. Són molt utilitzats en la indústria automobilística per les soldadures en arc, manipulació de materials, per assemblatge, etc.
- **Robots col·laboratius:** també coneguts amb el nom de Cobot, provinent de l'anglès *Collaborative Robots*, són robots articulats que permeten dur a terme tasques amb interacció segura amb humans en un espai de treball compartit.

Donat que aquest treball consta de la posada en marxa d'un robot col·laboratiu, en el següent apartat es profunditza sobre aquests tipus de robots.

### 2.3.1. Robots col·laboratius

Els robots col·laboratius són aquell que permeten una col·laboració directa entre humà-robot per realitzar tasques comunes. Segons el tipus d'interacció entre aquestes ambdues parts, es pot determinar el tipus de col·laboració.



**Figura 2.1.** Classificació de la col·laboració entre robot-operari (Font: [9])

Tal com es mostra en la figura anterior es defineixen quatre tipus d'operacions col·laboratives HRC (*Human-Robot Collaboration*). En la coexistència i la cooperació seqüencial no hi ha una col·laboració simultània, per tant, no és necessari un robot col·laboratiu donat que les tasques les pot fer un robot convencional/tradicional. En canvi, en la cooperació paral·lela i la col·laboració, l'humà i el robot treballen al mateix temps en un mateix lloc de treball. En aquest segon cas sí que és necessari aquest tipus de robot.

Segons el tipus de sistemes que incorporen, es poden distingir quatre modes d'operació en la robòtica col·laborativa [8]:

- 1- **Robot amb parada monitorada amb classificació de seguretat:** amb aquest sistema, el robot s'atura abans que l'operari pugui exposar-se a qualsevol perill en l'espai de treball col·laboratiu. Aquesta parada es du a terme quan l'operari entra a l'àrea de treball del robot. En nivell PL de seguretat correspon al "D" (informació més detallada en el capítol de seguretat). En aquest cas, el robot mai retornarà a la tasca de forma automàtica sense accionament per part de l'operari. Normalment és un robot convencional al qual incorpora un sensor per la detecció.

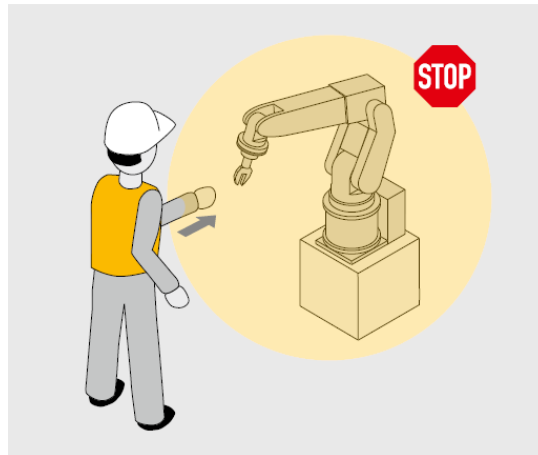


Figura 2.2. Representació de parada monitoritzada (Font: [10])

- 2- **Robot amb guiat manual:** amb aquest sistema l'operari utilitza un sistema d'accionament manual i el robot es mou segons les indicacions que efectua l'operari per guardar posicions o camins i facilitar en la programació.

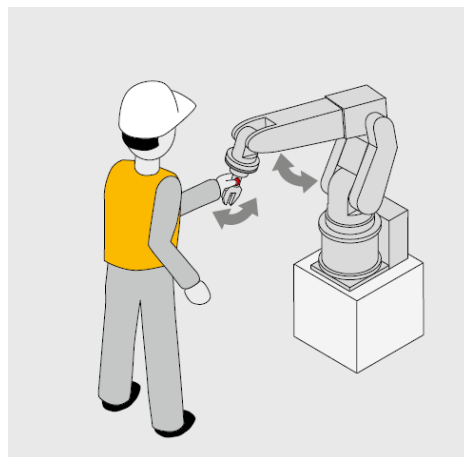
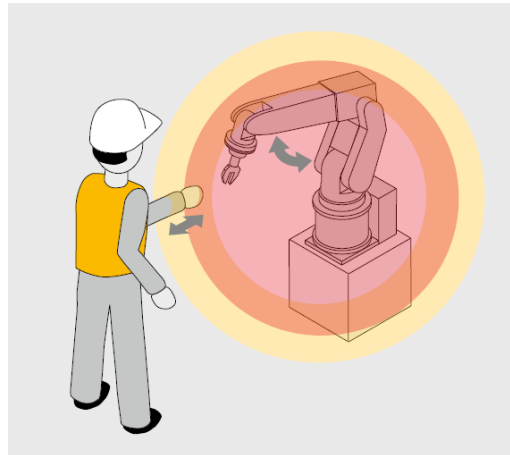


Figura 2.3. Representació guiat manual (Font: [10])

- 3- **Robot amb monitoratge de velocitat i separació:** aquests robots utilitzen dispositius de seguretat externs per detectar la posició i la seva distància. Es defineixen 3 zones, una de

treball en condicions normals, una zona d'advertència i una zona de perill. Si l'operari està en la zona de treball, el robot fa les seves tasques sense restriccions de velocitat, si entra en la zona intermèdia, el robot disminueix la velocitat a una de segura, si entra en la zona de perill, el robot s'atura de forma automàtica i reprèn l'activitat quan aquest abandona la zona. En el cas que es detecti que la distància entre humà-robot augmenta, el robot recuperarà l'estat òptim de treball.



**Figura 2.4.** Representació monitorització de velocitat i separació (Font: [10])

- 4- **Robot amb limitació de potència i força:** és el cas més comú donat que la majoria de robots col·laboratius estan dissenyats per limitar la potència i la força. En el cas que el robot detecta un cert nivell de força degut a una col·lisió, el robot es para com a mesura de seguretat. Aquest tipus de robots permeten treballar amb entorns sense necessitat de barreres de seguretat addicionals.

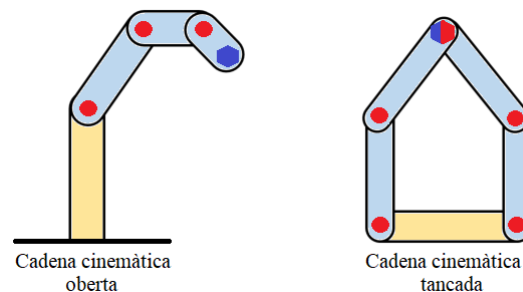


**Figura 2.5.** Representació limitació de potència i força (Font: [10])

## 2.4. Característiques bàsiques

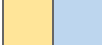



Les característiques que defineixen un robot industrial són:

- **Graus de llibertat (GdL):** s'entén per graus de llibertat d'una cadena cinemàtica (conjunt mecànic format per barres rígides unides per articulacions amb un extrem fixe i un mòbil) el conjunt de variables per definir la configuració de la cadena. Dit d'una altra manera, correspon al nombre d'actuadors per moure la cadena [11]. En el cas de robots amb cadena cinemàtica oberta (ex: robots articulats, UR3e) els graus de llibertat corresponen al nombre d'articulacions. En canvi, els robots amb cadena cinemàtica tancada (ex: robot dual-arm, Mitsubishi SCARA RP) tenen un nombre inferior de graus de llibertat.



**Figura 2.6.** Representació cadena cinemàtica oberta i tancada (Font: Bisnu Masó)

**Taula 2.1.** Paràmetres cadena cinemàtica.

Paràmetre	Color
Barres	
Articulacions	
Base	
Extrem	

Quan es treballen amb robots senzills és fàcil determinar els GdL analitzant el robot visualment però a mesura que augmenta la complexitat del robot, augmenta la dificultat per determinar els graus de llibertat. Pels casos més complexos s'utilitza la fórmula de Grübler:

$$GDL = \lambda \cdot (n - j - 1) + \sum_{i=1}^j f_i \quad (\text{Eq. 2.1})$$

On  $\lambda$  correspon a la dimensió de l'espai de treball (normalment 3 per 2D i 6 en el cas de 3D),  $n$  correspon al nombre de barres totals (incloent-hi la base),  $j$  correspon al nombre d'articulacions i  $f_i$  correspon als graus de llibertat de cada articulació.

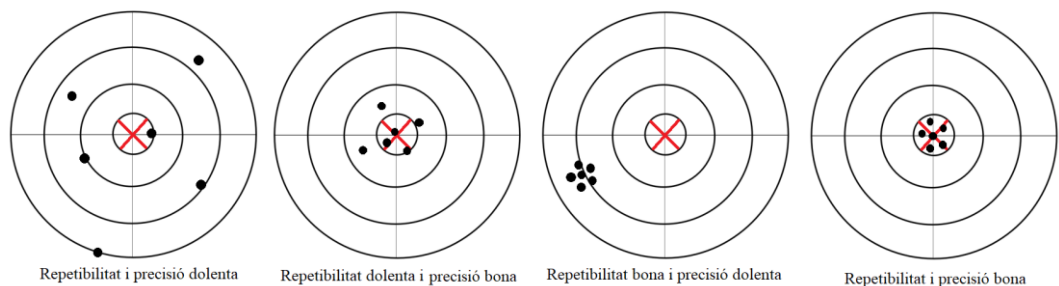


- **Àrea de treball:** correspon a un espai finit format per un conjunt de punts on el robot pot treballar, és a dir, aquells punts on l'extrem del robot pot arribar. Aquest factor varia segons les dimensions del robot, la seva constitució i els graus de moviment de les articulacions. En aquesta especificació tècnica no es té en compte l'element terminal donat que és responsabilitat del client i la seva aplicació.



**Figura 2.7.** Representació ares de treball d'un robot cartesià, scara i articulat (Font: [12])

- **Repetibilitat i precisió:** la repetibilitat és la capacitat d'un robot per repetir una mateixa tasca amb els mateixos moviments i recorreguts entre una tasca i una altra. La precisió és l'error entre la tasca assignada i el resultat de la tasca obtinguda [13], és a dir, la diferència entre la posició assignada per l'usuari i la posició obtinguda pel robot. Aquests dos termes són els que es poden mesurar de forma més senzilla i alhora són confosos entre ells. A la següent figura es pot veure una representació gràfica de cada concepte:



**Figura 2.8.** Tipus de repetibilitat i precisió (Font: Bisnu Masó)

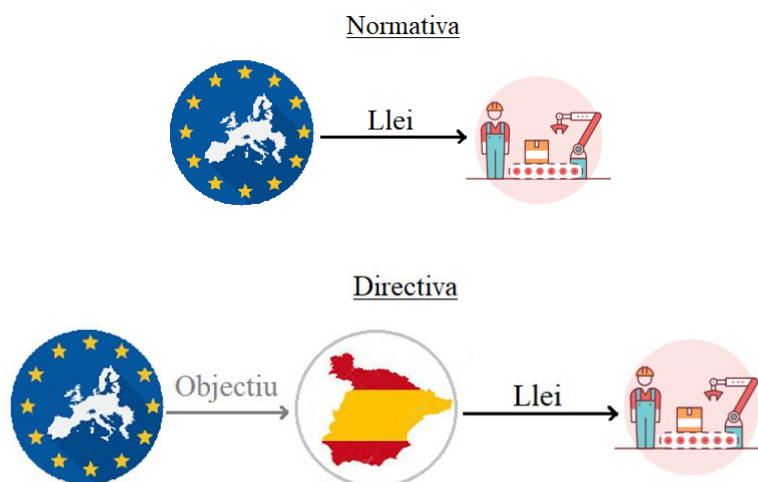
- **Resolució:** la resolució del robot ve determinat pel canvi o moviment més petit que pot dur a terme un actuator o pot ser detectat per un sensor de retroalimentació (dispositiu que proporciona al sistema de controls mesures de quantitats físiques per tancar el llaç del sistema de control de l'actuator [14]).
- **Capacitat de càrrega:** és el pes màxim que un robot pot aixecar per treballar en condicions òptimes sense alterar-se la precisió [15]. S'ha de tenir en compte que aquesta càrrega màxima varia segons la posició de treball en l'interior de la zona de treball. En el cas dels robots

articulats, aquesta càrrega màxima disminueix a mesura que s'acosta al límit exterior de l'àrea de treball. Pot alterar-se per efecte de la gravetat, la temperatura, l'acceleració, etc.

- **Velocitat i acceleració:** el primer concepte fa referència a la velocitat màxima que el robot pot assolir en els seus moviments respecte del TCP (de l'anglès *Terminal Center Point*) o les articulacions individuals. L'acceleració fa referència a l'increment màxim de la velocitat per unitat de temps fins a assolir la velocitat màxima que poden efectuar els actuadors del robot.

## 2.5. Legislació

Segons l'àmbit d'aplicació de la legislació es poden diferenciar en dos blocs: les normatives o reglaments i les directives. Els reglaments són actes legislatius que tenen fonament jurídic que s'han d'aplicar en tots els estats de la Unió Europea i és comú en la seva integritat. En canvi, les directives són actes legislatius en els quals s'estableixen objectius que determinats o tots els estats membres de la UE han de complir però cada país elabora les seves pròpies lleis per assolir aquests objectius. A la següent figura es mostra una representació gràfica per diferenciar aquests dos conceptes.



**Figura 2.9.** Diferència entre les normatives i les directives en format gràfica (Font: Bisnu Masó)

En el cas de l'estat espanyol, l'organització encarregada d'adaptar els objectius i fer lleis és l'AENOR (*Asociación Española de Normalización y Certificación*). En l'àmbit de la robòtica, aquestes normatives es classifiquen en dos grups. En els casos dels robots industrials han de complir la UNE-EN ISO 10218-1:2012 que porta el títol: *Robots y dispositivos robóticos. Requisitos de seguridad para robots Industriales* i també la UNE-EN ISO 10218-2:2012 amb el títol: *Robots y dispositivos robóticos.*

Requisitos de seguridad para robots industriales. Parte 2: Sistemas robot e integración. (ISO 10218-2:2011). [16]

En el cas de robots no industrials han de complir la UNE-EN ISO 13482 amb el títol: Robots y dispositivos robóticos. Requisitos de seguridad para robots no industriales. Robots de asistencia personal no médicos.

## 2.6. Seguretat

La seguretat en la indústria actual es pot dividir en dues tipologies de seguretat: la seguretat intrínseca i l'extrínseca.

- **Seguretat intrínseca:** és aquella seguretat que ha de tenir el dispositiu. La responsabilitat per complir aquests reglaments recau sobre el fabricant i l'integrador queda absent del seu compliment. Aquesta seguretat fa referència a les normatives ISO 10218-1, EN ISO 12100, EN 60204-1, directiva de màquines, etc.
- **Seguretat extrínseca:** són els mètodes de seguretat que ha d'implementar el client per a poder utilitzar aquest dispositiu en l'entorn que es desitgi. Aquest tipus de seguretat depèn de l'entorn de treball on es vulgui integrar el dispositiu. Les normatives que fan referència a aquest tipus de seguretat són l'ISO 10218-2, EN ISO 14120, EN ISO 13857, EN ISO 14119, etc.

Una de les principals legislacions que estableix els requisits principals de salut i seguretat en les màquines, a nivell de la Unió Europea, és la Directiva de Màquines 2006/42 / CE que substitueix a l'antiga Directiva 98/37/CE [17]. Aquesta directiva determina que les màquines no han de representar un risc per a les persones que treballen en una zona industrial. Al ser una directiva, els aspectes anteriors són objectius a complir i cada estat ha d'implementar la seva normativa. Per arribar a l'objectiu, existeixen dues normatives:

- **EN ISO 13849-1:** Components dels sistemes de control relatius a la seguretat, principis generals de disseny. En aquesta normativa, es determina el nivell de fiabilitat necessari (PL, Performance Level).

**Taula 2.2.** Nivell de fiabilitat i la seva probabilitat [18].

Performance Level (PL)	Probability of Dangerous Failure per Hour (PFHd) 1/h
<b>A</b>	$\geq 10^{-5}$ and $< 10^{-4}$ $\langle 0.001\% \text{ to } 0.01\% \rangle$
<b>B</b>	$\geq 3 \times 10^{-6}$ and $< 10^{-5}$ $\langle 0.0003\% \text{ to } 0.001\% \rangle$
<b>C</b>	$\geq 10^{-6}$ and $< 3 \times 10^{-6}$ $\langle 0.0001\% \text{ to } 0.0003\% \rangle$
<b>D</b>	$\geq 10^{-7}$ and $< 10^{-6}$ $\langle 0.00001\% \text{ to } 0.0001\% \rangle$
<b>E</b>	$\geq 10^{-8}$ and $< 10^{-7}$ $\langle 0.000001\% \text{ to } 0.00001\% \rangle$

- **EN 62061:** Normativa relativa a la seguretat de sistemes de controls relatius referents a la seguretat elèctrica, electrònica i programables. En aquesta normativa es dur a terme una avaluació de riscos i definició del nivell d'integritat de la seguretat necessària (SIL, Safety Integrity Level )

**Taula 2.3.** Nivell d'integritat de seguretat [19].

Safety Integrity Level	Probability of Failure on Demand	Factor de reducció de risc
<b>SIL 4</b>	$\geq 10^5$ to $< 10^4$	100,000 to 10,000
<b>SIL 3</b>	$\geq 10^4$ to $< 10^3$	10,000 to 1,000
<b>SIL 2</b>	$\geq 10^3$ to $< 10^2$	1,000 to 100
<b>SIL 1</b>	$\geq 10^2$ to $< 10^1$	100 to 10

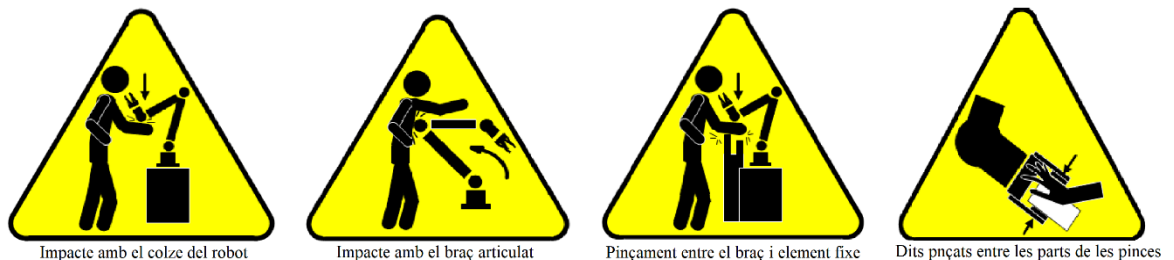
Tal com es pot observar, la normativa ISO és més estricta en tenir més nivells de seguretat PL (a, b, c, d, e) envers la normativa EN 62061 que només en té tres (SIL1, SIL2, SIL3) [20].

Respecte a les normatives, la normativa ISO 10218 fa referència a tot el conjunt de normes i directrius que han de complir els robots en l'actualitat. Aquesta normativa està dividida en dues parts: l'ISO 10218-1 i la ISO10218-2. La primera part parla sobre les característiques de seguretat que han de complir tots els robots perquè puguin ser utilitzats en una fàbrica. Aquesta normativa l'ha de complir el fabricant del robot. Per obtenir una màquina segura s'ha de realitzar una avaluació de riscos a partir de la normativa UNE-EN ISO 12100 *Seguridad de las máquinas. Principios generales para el diseño. Evaluación del riesgo y reducción del riesgo* (ISO 12100:2010) en la fase del disseny i construcció. La segona part, explica tot el conjunt de normes que s'han de tenir en compte a l'hora d'integrar un robot en un procés industrial. Aquesta normativa l'ha de complir l'integrador del robot.

En el cas d'existir un terme de seguretat en ambdós documents, prevaldrà la que suposi més seguretat. Per exemple: si aplicant una directriu permet tenir un nivell de seguretat superior que aplicant una normativa, la directriu prevaldrà per sobre la normativa.

### 2.6.1. Seguretat robots col·laboratius

Amb l'aparició dels robots col·laboratius han donat lloc a unes noves tasques on hi ha una interacció entre humà i robot (HRC) en el mateix lloc de treball i de forma simultània. Per aquest motiu apareixen nous riscos que no estaven contemplats amb els robots tradicionals donat que l'única interacció es produïa quan el robot estava aturat. En la següent figura es poden veure alguns dels riscos que es poden produir ja sigui per moviments inesperats del robot, error en el límit de la força, error en el límit de la velocitat, desviació respecte a la trajectòria programada, etc.



**Figura 2.10.** Nous riscos que comporten els robots col·laboratius (Font: [21])

Donat que no existia una normativa de seguretat específica per als robots col·laboratius ni en l'ISO 10.218-1 ni en l'ISO 10.218-2, es va elaborar l'ISO / TS 15066: 2016 que porta el títol: Robots and robotic devices — Collaborative robots on s'especifiquen els requisits de seguretat per als sistemes de robot industrial col·laboratiu i l'entorn de treball, i complementa els requisits i l'orientació sobre el funcionament del robot industrial col·laboratiu que figuren en ISO 10.218-X. Aquesta normativa s'aplica als sistemes de robots industrials i no s'aplica als robots no industrials, tot i que els principis de seguretat presentats poden ser útils per a altres àrees de la robòtica [22].

### 3. Elecció del robot

#### 3.1. Estat de l'art dels robots col·laboratius

En l'actualitat hi ha un mercat bastant ampli de robots col·laboratiu donat que la demanda d'aquests robots està en ple creixement pel fet de poder-se utilitzar per a tasques o espais on abans no era possible incorporar-hi un robot tradicional. Els principals fabricants que es poden trobar són: Universal Robots (UR16e), Kuka (LBR iiwa), FANUC (CR-35iA), ABB (Yumi IRB), i Rethink Robotics (Sawyer) [23]. En la següent taula es mostra una comparativa dels diferents robots que ofereix cada empresa.

**Taula 3.1.** Comparativa robots col·laboratius.

Robot	GdL	Àrea treball (mm)	Velocitat max TCP (m/s)	Càrrega max (kg)	Repetibilitat (mm)
UR16e	6	900	1	16	± 0,05
LBR iiwa	7	820	2,36 rad/s	14	± 0,15
CR-35iA	6	1813	0,75	35	± 0,04
Yumi IRB	6	559	1,5	0,5	± 0,02
Sawyer	7	1260	1,5	4	± 0,1

**Nota:** (El criteri per escollir el robot es dur a terme a partir d'un concurs públic de la UPC on es determinen unes especificacions que han de complir els robots que ofereixen les empreses que entren en el concurs. També es tenen en compte diferents aspectes com els nivells d'aprenentatge que ofereix el robot, la inserció del robot en la indústria actual i el preu).

Dels fabricants anteriors es va optar pel model d'Universal Robots pels avantatges que ofereix tant a nivell d'especificacions (apartat 3.2.1), els recursos i la facilitat d'aprenentatge (oferint programació de nivell bàsic fins a nivell avançat) i la seva gran inserció en la indústria que permet potenciar les sortides professionals dels estudiants.

#### 3.2. Universal Robots

Universal Robots, amb origen a Dinamarca, es considerada una de les empreses punteres en l'elaboració i distribució de robots col·laboratius en l'actualitat. L'empresa es va crear el 2005 per tres enginyers de la Universitat de Dinamarca amb l'ajuda de la inversió de la companyia Syddansk Innovation, empresa que es dedica al capital de risc invertint en empreses tecnològiques emergents (*startups*). Al cap de tres anys, el 2008 van treure al mercat el primer robot col·laboratiu, el model UR5, amb una capacitat per aixecar pesos de 5 kg. Més endavant, el 2012 varen treure al mercat la versió

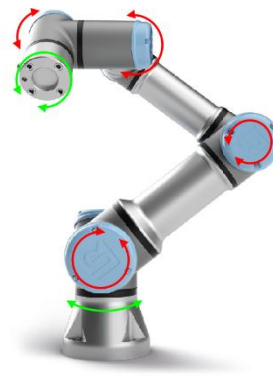
UR10 capaç d'aixecar el doble de pes que el model anterior amb una àrea de treball de 130 cm de radi. El 2014 va tenir un creixement molt important que va suposar un augment del 70% respecte a l'any anterior. El següent any la companyia fou comprada per la companyia Teradyne Inc. El 2016 van treure al mercat el sistema UR+ que permet a empreses externes fabricar complements (elements terminals, softwares...) compatibles amb els robots UR de manera oficial. D'aquesta manera, la companyia es centra, exclusivament, en el desenvolupament i fabricació de robots. Dos anys més endavant, el 2018, treuen al mercat la gamma e-series (UR3e, UR5e i UR10e), amb millores en les prestacions, tant a nivell de hardware com a software. Finalment, el 2019, la companyia ja havia venut més de 35000 robots arreu del món. En l'actualitat, l'empresa Universal Robots, opera en 20 països, té una quota de mercat del 50 % en els braços robòtics col·laboratius, té 400 empreses adherides al programa UR+ i ha tret un nou producte de la gamma e-series, l'UR16e [24].

### 3.2.1. UR3e - Especificacions

El model UR3e és el model més petit que ofereix el fabricant però les seves especificacions fan que sigui el robot ideal per la incorporació d'un robot col·laboratiu al laboratori de robòtica de la universitat.

Les característiques són les següents:

- **Graus de llibertat (GdL):** per determinar els graus de llibertat es poden utilitzar un dels dos mètodes següents:
  - Mètode 1: donat que el robot és un robot articulat, els GdL correspon al nombre total d'actuadors que aquest conté. Per tant, com que conté 6 actuadors, es pot determinar que el robot té 6 graus de llibertat.

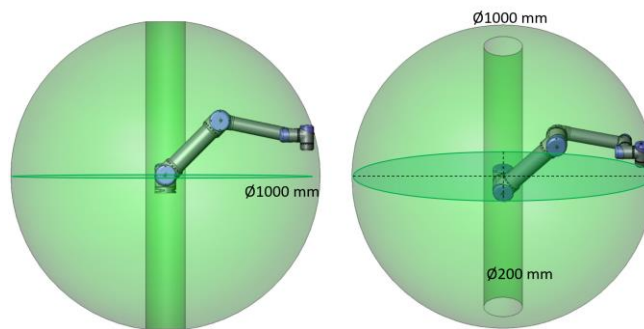


**Figura 3.1.** Determinació graus de llibertat robot UR3e (Font: Bisnu Masó)

- Mètode 2: aplicant la fórmula de Grübler (equació 4.1):

$$\text{GDL} = 6 \cdot (7 - 6 - 1) + \sum_{i=1}^6 1_i = 6 \quad (\text{Eq. 3.1})$$

- **Àrea de treball:** els robots d'Universal Robots estan construïts amb articulacions que permeten girs de volta completa, és a dir, de 360°. Al no haver-hi limitacions de gir, l'àrea de treball del robot és bastant gran i permet arribar a una distància de radi 50 cm formant una esfera respecte al punt mig de la base. Indiferentment de la manera i lloc que estigui col·locat (taula, paret vertical, sostre, pla inclinat...) l'únic impediment seran l'existència de plans que no pot travessar: taula, paret, zones de seguretat... També s'ha de tenir en compte que les posicions límit del robot (quan aquest està completament estirat) no són recomanables per dur a terme tasques de manera eficaç donat que el moviment de les articulacions es duen a terme a grans acceleracions.

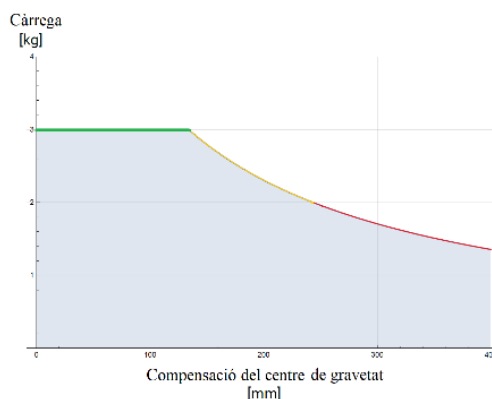


**Figura 3.2.** Àrea de treball robot UR3e (Font: [1])

- **Repetibilitat:** la normativa UNE-EN ISO 9283:2003, que porta com a títol *Robots manipuladores industriales. Criterios de análisis de prestaciones y métodos de ensayo relacionados. (ISO 9283:1996)*, determina les prestacions mínimes que ha de tenir el robot i com obtenir aquestes mesures que determina si un robot compleix la normativa de robots manipuladors. La repetibilitat d'aquest model escollit és de  $\pm 0,03$  mm amb càrrega i està contemplada en la normativa anterior.
- **Resolució:** (la resolució dels actuadors no s'especifica).
- **Capacitat de càrrega:** el pes màxim capaç de manipular és de 3 kg. Donat que el centre de gravetat de l'element termina no sempre és centrat amb el punt de connexió amb el robot, aquesta càrrega útil varia segons la compensació del centre de gravetat (distància entre el centre de l'anella de connexió i el centre de gravetat de l'element terminal) donat que, una càrrega amb el centre de gravetat molt allunyat del punt de connexió, actuaria com una



palanca amplificant la força que ha de suportar l'extrem del robot fins al punt que el podria malmetre.



**Figura 3.3.** Gràfica capacitat de càrrega respecte el centre de gravetat (Font: [1])

- **Velocitat:** tal com s'ha comentat anteriorment, cinc de les 6 articulacions tenen una llibertat de gir de  $\pm 360^\circ$  i el que falta, corresponent a la peça d'unió amb l'element terminal, no té límit de gir i pot girar de manera lliure. Les tres primeres articulacions, començant per la base, poden girar a una velocitat màxima de  $180^\circ/\text{s}$  i les altres tres restant, poden anar a una velocitat màxima de  $360^\circ/\text{s}$ . Per altra banda, la velocitat típica respecte al TCP és d'1 m/s.
- **Nivell de seguretat:** incorpora un sistema avançat de control de força en cada un dels eixos que permet ajustar-ho entre 100 i 250 newtons, compleix la normativa UNE-EN ISO 13849-1:2015 amb la categoria Cat.3, nivell de fiabilitat PL del tipus d, també compleix la normativa UNE-EN ISO 10218-1 i disposa d'un sistema de seguretat, UR Safety 3.0 amb certificat TÜV, amb el qual es permet configurar la força, la velocitat, la potència i el moment de les articulacions [25].

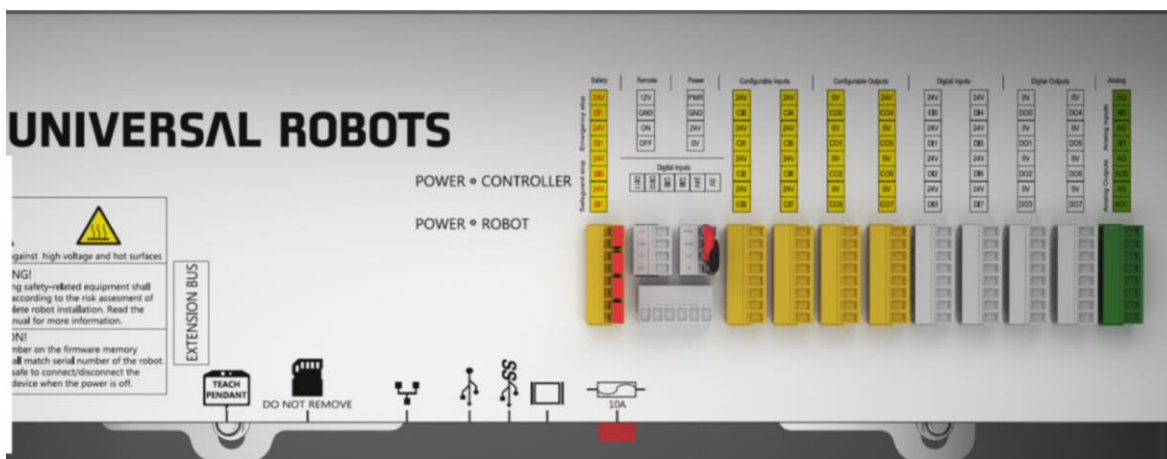
**Taula 3.2.** Resum característiques UR3e.

Característica	Descripció
Graus de Llibertat	6
Àrea de treball (mm)	$\varnothing 1000$
Repetibilitat (mm) (m/s)	$\pm 0,03$
Càrrega max (kg)	3
Velocitat max TCP	1 m/s (típica)
Seguretat	UNE-EN ISO 13849-1:2015, UNE-EN ISO 10218-1, CAT 3 i PL D.

### 3.2.2. Ports d'E/S

Els ports d'entrada/sortida i els ports de comunicacions que disposen es poden trobar en dos llocs diferents. La gran majoria estan situats en la caixa de control però també se'n poden trobar en l'extrem del braç articulat en el connector Lumberg de 8 pins.

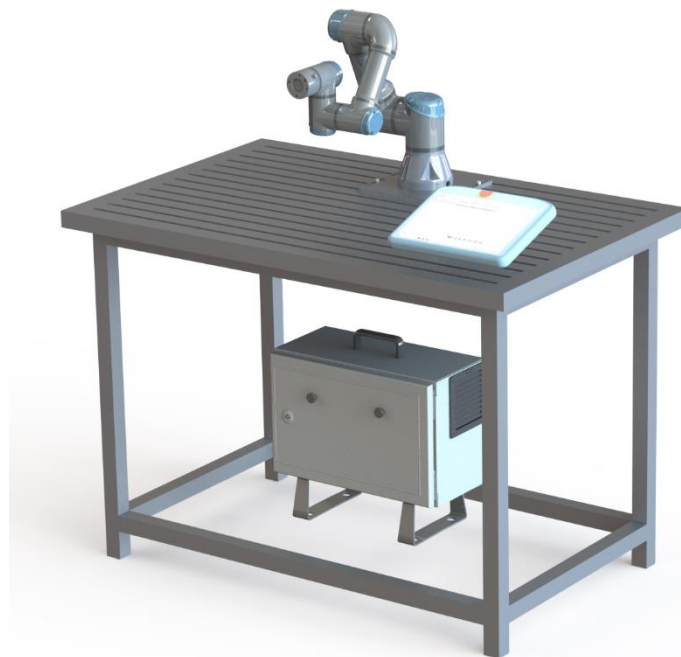
En la caixa de control, tal com es mostra en la figura 3.4, hi podem trobar els pins de connexió d'entrades i sortides i connexions d'elements de seguretat i accionament. Tots ells treballen amb una tensió de 24 V. La primera fila situada a la part més esquerra, els connectors de color groc, permeten connectar-hi botons de parada d'emergència que funcionen paral·lelament amb el botó de parada d'emergència que hi ha de teach pendant o consola de programació. Els connectors situats a la seva dreta permeten connectar-hi botons o altres sistemes, com per exemple un PLC, per encendre/apagar el robot sense fer ús del botó de la consola de programació. Per altra banda, els que estan situats de forma horitzontal, són ports utilitzats per connectar-hi una cinta de transport i poder sincronitzar el moviment d'aquest amb el robot i permetre un treball síncron sense necessitat de parar la cinta. Les següents quatre files de connectors de color groc, permeten connectar dispositius de seguretat que actuïn com a entrades de senyals (4 dispositius) o sortides (4 dispositius), per exemple: làsers de seguretat, plataformes sensibles, bloquejadors de portes... Els següents connectors de color gris corresponen a pins de connexió de senyals digitals, 16 pins d'entrada i 16 pins de sortida (8 dispositius per a cada tipus). Per últim, el connector de color verd permet connectar-hi dispositius analògics, en aquest cas conté 2 entrades i 2 sortides analògiques.



**Figura 3.4.** Interior controlador robot UR3e i els pots de connexió (Font: [2])

En la part inferior de la caixa que conté els dispositius de control, s'hi poden trobar (d'esquerra a dreta): el connector per la consola de programació, una ranura per la targeta de memòria que conté el sistema operatiu, port de connexió Ethernet RJ-45, dos ports de connexió USB (USB2.0 i USB3.0) i un port de connexió mini DisplayPort.

Respecte al connector Lumberg, model KKMV 8-354, que es troba en la brida per connectar-hi l'element terminal, conté 8 pins que corresponen a: un pin de terra (GND), un pin d'alimentació que es pot seleccionar entre 0V, 12V o 24V, dos pins corresponents a la sortida digital DO0 i DO1, dos pins corresponents a les entrades digitals DI0 i DI1, dos pins corresponent a entrades analògiques en AI2 o AI3. Els pins de les entrades analògiques també es poden utilitzar per establir una comunicació a través del protocol RS-485. El conjunt d'aquestes connexions estan dissenyades per establir una comunicació directa entre l'element terminal i el robot i facilitar la lectura dels diferents sensors o dades que aquest pugui transmetre.



**Figura 3.5.** Estació bàsica amb el braç robòtic, la consola de programació i la caixa del controlador (Font: Bisnu Masó)

### 3.2.3. Comunicacions

A partir del port de connexió Ethernet es possible establir quatre tipus de comunicacions: Modbus TCP, EtherNet/IP, Profinet i Ethernet Socket. A continuació es defineixen de forma molt breu donat que no són els conceptes d'estudi, només la seva comprensió:

- **MODBUS TCP:** és un de protocol de comunicació dissenyat específicament per aplicacions industrials destinat a la supervisió i control d'equips d'automatització que permet una comunicació en l'entorn d'Internet utilitzant l'arquitectura client/servidor, el protocol TCP/IP [26]. Permet la comunicació de diferents dispositius d'una mateixa xarxa, els quals posseeixen una adreça IP única, i ha esdevingut un protocol molt integrat per la indústria tot i no haver passat per cap organisme d'estandardització. El tipus de dades que és capaç de distingir són:

entrades digitals (*discret input*), sortides digitals (*coils*), registres d'entrada (*input register*) i registres de retenció (*holding registers*).

- **EtherNet/IP**: és un protocol de xarxa industrial que adapta el CIP (Common Industrial Protocol) a l'estàndard Ethernet i utilitza els dos dels estàndards més implementats: el conjunt de protocols d'internet i el IEEE 802.3. Tanmateix, també ofereix transferència de dades en temps real i permet la transferència de dades bàsiques d'entrades i sortides a través de missatges implícits (port 2222), carregar i descarregar paràmetres a través de missatges explícits (port 44818), comunicació a un dispositiu o varis a la vegada, etc.
- **PROFINET**: de l'anglès *Process Field Network*, és un estàndard industrial (IEC 61784-2) desenvolupat per PROFIBUS International per l'intercanvi d'informació a través de la xarxa Ethernet per la lectura de dades i control de diferents dispositius. Creat per satisfer la necessitat d'intercanvi de dades de manera ràpida i en temps real (RT) que exigia la indústria automàtica, assolint temps de cicle inferiors a 31,25 µs [27]. El concepte de PROFINET es pot descriure a partir de la següent equació:

$$PROFINet = PROFIBUS + x + y + z [28] \quad (\text{Eq. 3.2})$$

x→ Obrir una comunicació constant amb PROFIBUS i l'estàndard informàtic Ethernet.

y→ Model d'enginyeria independent del fabricant.

z→ Distribució d'automatització mitjançant un model d'objectes oberts.

- **Ethernet Sockets**: la comunicació per sockets, on també s'utilitza el protocol TCP/IP, és un mecanisme de comunicació bidireccional entre dos dispositius o dos processos d'un mateix programa basada en la filosofia client-servidor (el client fa la petició del servei i el servidor l'hi envia). A diferència de les comunicacions anteriors, no existeix node de connexió donat que només permet la comunicació directa entre dos dispositius.

Tant MODBUS TCP com PROFINET permeten la transferència de dades en temps real però no permeten modificacions de paràmetres amb missatges explícits com la comunicació Ethernet Sockets que, en contra, no permet la transferència en temps real. En canvi EtehNet/IP és un protocol que ofereix les dues possibilitats. A l'hora de triar la comunicació s'ha de tenir en quants equips es volen connectar alhora (per exemple Ethernet Socket només permet connectar-se entre dos dispositius) i els protocols que disposen la resta de dispositius perquè no són compatibles entre ells.

De les comunicacions anomenades anteriorment s'ha escollit treballar amb la comunicació Ethernet Sockets perquè és l'únic que ens ha permès establir una comunicació funcional entre el simulador del

robot i el software Matlab. Les altres comunicacions no s'han pogut testejar amb el robot físic i amb el simulador no s'han obtingut resultats satisfactoris.

### 3.2.3.1. Comunicació per Sockets

El tipus de socket utilitzat és el socket de flux que, una vegada es realitza la connexió, es busca un camí lliure entre origen i destí i aquest es manté en tota la connexió [29].

Amb la comunicació bàsica per Socket permet controlar el robot des d'un dispositiu extern sense necessitat d'un programa que s'executi al robot. Aquesta connexió es pot dur a terme a través dels ports 30001, 30002 o 30003. Per a altra banda també permet tenir un programa al robot, el qual actua com a client, i intercanviar dades amb un altre dispositiu extern que actuï com a servidor per augmentar les prestacions [30]. Per aquest tipus de comunicació s'aconsella utilitzar un port diferent dels esmentats anteriorment per no haver-hi conflictes entre client-servidor. Tant si s'utilitza un port com un altre, la programació es desenvolupa a través de l'enviament de seqüències de comandes sense format [31] que són interpretades i executades pel controlador del robot.

A continuació es detalla l'estructura de la part de comunicació del programa que s'executa en el robot.

L'estructura principal del programa consta de tres parts bàsiques:

- **BeforeStart:** s'hi determinen aquelles accions que s'han de dur a terme abans d'iniciar el programa del robot. En aquest cas l'obertura de la comunicació. El bucle s'utilitza per establir la connexió si en el primer camí escollit no s'estableix la connexió.
- **Robot Program:** en aquest apartat s'inclouen tots els codis que fan referència a les accions del robot, acció segons els valors rebuts per socket i la definició de moviments i trajectòries.
- **Thread\_1:** Conjunt de comandes que s'executen paral·lelament amb el *Robot Program*. El codi que s'inclou en aquesta part del programa és el que correspon a la lectura contínua de la informació que es rep del servidor. El bucle d'aquesta part s'utilitza per detectar si hi ha errors en la recepció i poder tornar a llegir les dades que es reben.

En el cas del programa pel robot UR té la següent estructura:

El número entre parèntesis després de *socket\_read\_ascii\_float(x)* indica que només agafi un valor de lectura. Les línies de codi del *Thread\_1* s'utilitzen per rebre i llegir dades, les del *Thread\_2* s'utilitzen per enviar informació pel Socket.

```

Program
BeforeStart
  'Establiment de la connexió entre client i servidor.'
  open:=socket_open("IP ROBOT",PORT)
  Loop open= False
    Wait: 0.01
    open:=socket_open("IP ROBOT",PORT)
  'Posició inicial de la targeta'
  PosicioInicial:=p[0,0,0,0,0,0]

Robot Program
  'Programa d'execució de tasques del robot.'

Thread_1
  'Lectura de dades enviades pel servidor.'
  dadaServer:=socket_read_ascii_float(1)
  Loop dadaServer[0]≠1
    Wait: 0.1
    dadaServer:=socket_read_ascii_float(1)

Thread_2
  'Enviament dades cap al servidor.'
  socket_send_line("0")

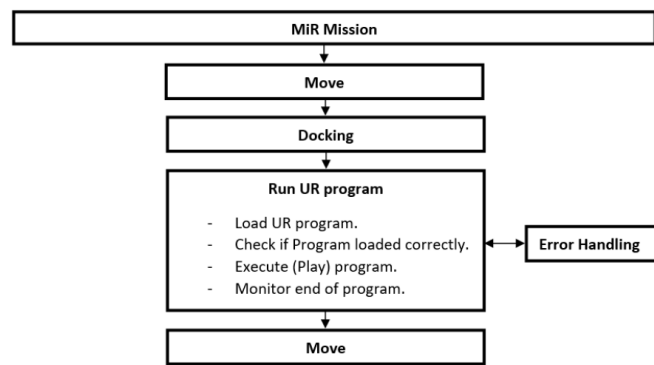
```

Un exemple real d'ús d'aquest protocol de comunicació és amb la plataforma mòbil del fabricant MIR (*Mobile Industrial Robots*) en els quals hi incorporen un robot UR i utilitzen la comunicació per Socket per controlar el braç articulat. En aquest cas, la plataforma fa de client i fan ús de les funcions de *Dashboard Server* del robot per enviar-li instruccions implícites (carregar programa, executar el programa i monitoratge).



**Figura 3.6.** Plataforma MIR100 amb robot UR5e (Font: [32])

En el següent diagrama es mostra la seqüència que es dur a terme quan els dos dispositius treballen conjuntament. Una vegada la plataforma mòbil ha arribat a la seva destinació, envia les ordres al robot, aquest les executa i una vegada finalitzades, es torna a iniciar el moviment la plataforma.



**Figura 3.7.** Diagrama de flux de la “missió” del robot MIR (Font:[32])

### 3.2.4. Llenguatge de programació

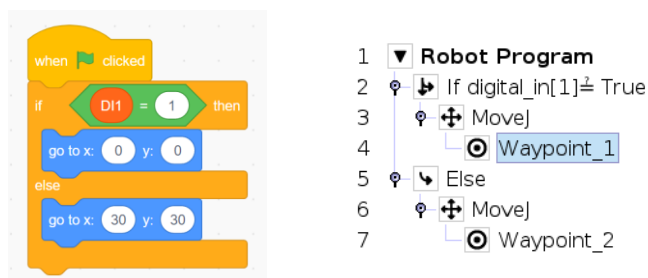
Pel que fa al llenguatge de programació en la robòtica no existeix cap estàndard que normalitzi o reguli aquest concepte. Per aquesta raó existeixen diversos llenguatges de programació en la robòtica. Els llenguatges genèrics més comuns que es poden trobar són: el pascal que va ser un dels primers llenguatges utilitzats, l'Scratch que s'utilitza en la robòtica educativa, Matlab per anàlisis de dades i desenvolupament de sistemes de control, HDL (Hardware Description Language), Python que està en creixement en l'actualitat, Java i C/C++ que és el llenguatge més utilitzat en la robòtica. Per altra banda es troben els llenguatges que desenvolupa cada fabricant de robots. De les marques més comunes es tenen que: ABB utilitza el llenguatge Rapid, Kuka té el llenguatge KRL, Jaskawa utilitza INFORM, Kawasaki utilitza AS, Fanuc utilitza Karel i Universal Robots utilitza URScript.

Donat que existeixen una quantitat elevada de llenguatges, amb l'aparició de la programació offline i la programació guiada es busca un llenguatge d'ús general que permeti trobar un “estàndard”. Una de les tendències actuals és el llenguatge de programació ROS-industrial [33]. Tot i així, és complicat determinar quin és el millor llenguatge de programació.

Tal com s'ha comentat anteriorment la marca Universal Robot aposta pel llenguatge URScript i ofereix tres formes de programació amb aquest llenguatge. Dues d'aquestes formes es diferencien per la forma de programar i la seva complexitat, la tercera és una combinació d'aquestes dues on a partir de la programació intuïtiva permet cridar funcions programades a través de scripts [31].

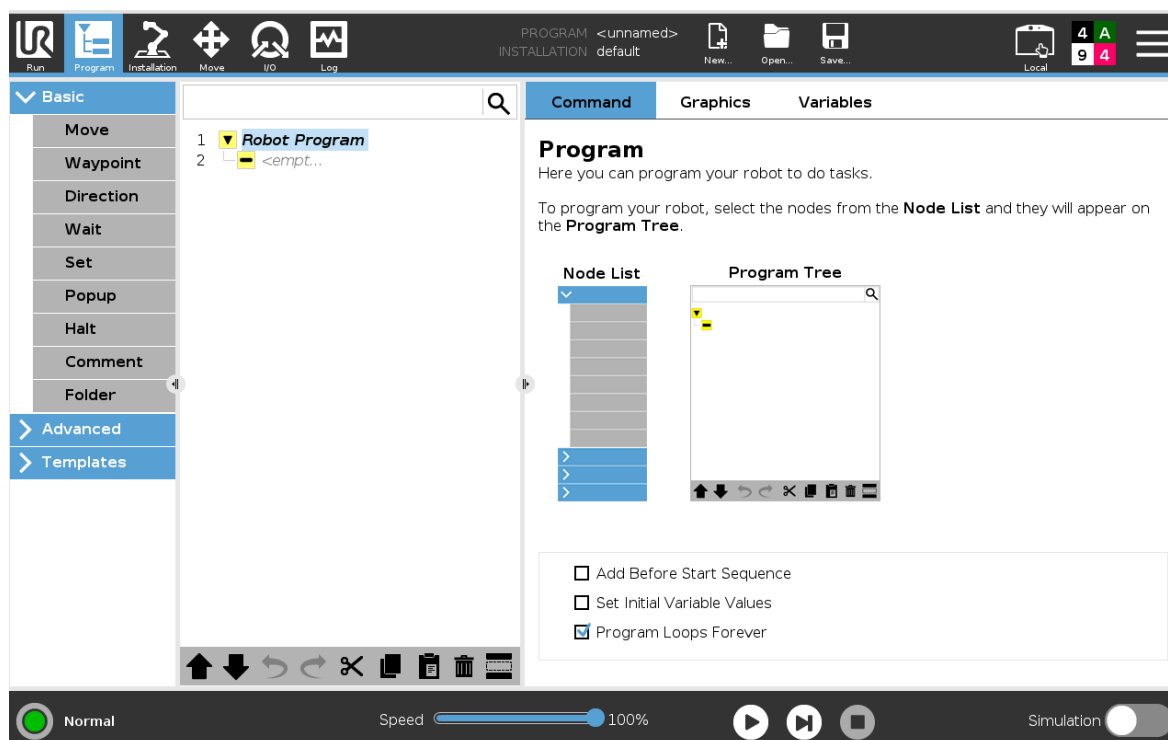
### 3.2.4.1. Programació intuïtiva

El primer mètode que permet programar el robot és la programació intuïtiva i aquesta es dur a terme a partir de la consola de programació del robot. Tal com es pot veure en la següent figura, aquesta té una semblança molt propera al llenguatge de programació LOGO o al llenguatge de programació Scratch, un llenguatge utilitzat per ensenyar a programar als infants a través del joc i l'experimentació.



**Figura 3.8.** A l'esquerra programació Scratch, a la dreta programació intuïtiva UR (Font: Bisnu Masó)

El mètode de programació que s'utilitza en el PolyScope està basat en blocs (en aquest cas blocs de funcions textuals) i un interfície molt senzilla i intuïtiva.

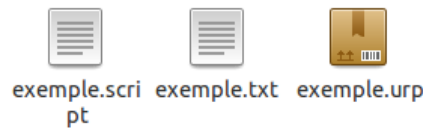


**Figura 3.9.** Captura pantalla de programació UR (Font: Bisnu Masó)

Una vegada es guarda el programa creat a partir del PolyScope, es generen tres fitxers relacionats amb el programa: un fitxer xxxx.urp que correspon al fitxer principal del programa, un xxxx.txt que és un



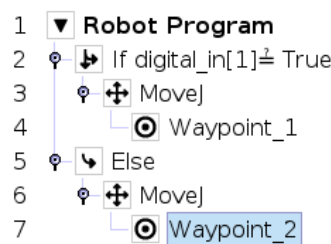
fitxer de text simple que il·lustra la seqüència del programa i, finalment, un xxxx.script que correspon al programa del robot amb llenguatge de programació URScripts.



**Figura 3.10.** Fitxers generats amb la programació intuïtiva (Font: Bisnu Masó)

### 3.2.4.2. Programació per Scripts

El segon mètode de programació és a partir del **llenguatge de programació URScripts** a partir d'escriptura directa que, a diferència de la programació intuïtiva, permet crear programes més complexos i avançats. En la figura següent consta de l'exemple del programa amb llenguatge de programació intuïtiu anomenat **Exemple** i, en la següent taula, el programa amb llenguatge de programació UR Script generat automàticament.



**Figura 3.11.** Exemple seqüència programació intuïtiva (Font: Bisnu Masó)

```
def exemple():
    set_standard_analog_input_domain(0, 1)
    set_standard_analog_input_domain(1, 1)
    set_tool_analog_input_domain(0, 1)
    set_tool_analog_input_domain(1, 1)
    set_analog_outputdomain(0, 0)
    set_analog_outputdomain(1, 0)
    set_input_actions_to_default()
    set_safety_mode_transition_hardness(1)
    set_gravity([0.0, 0.0, 9.82])
    set_tcp(p[0.0,0.0,0.0,0.0,0.0,0.0])
    set_payload(0.0)
    set_tool_communication(False, 115200, 0, 1, 1.5, 3.5)
    set_tool_output_mode(0)
    set_tool_digital_output_mode(0, 1)
    set_tool_digital_output_mode(1, 1)
    set_tool_voltage(0)
    global Waypoint_1_p=p[-.136861537556, -.267180821730,
    .374690844449, -.001221359682, 3.116276528482, .038891915637]
```

```

global Waypoint_1_q=[-1.6006999998292395, -1.5222631860476419,
-1.265790142741583, -1.9500466712066231, 1.5951000000043865, -
0.030999999829245617]
global Waypoint_2_p=p[-.136861537556, -.267180821725,
.193949177784, -.001221359682, 3.116276528482, .038891915637]
global Waypoint_2_q=[-1.600699999964771, -1.5868096749514624,
-2.081542373300593, -1.0697479517470816, 1.5951000000009028, -
0.03099999996477365]
while (True):
    $ 1 "Robot Program"
    $ 2 "If digital_in[1]≠ True "
    if (get_standard_digital_in(1) == True ):
        $ 3 "MoveJ"
        $ 4 "Waypoint_1"
        movej(get_inverse_kin(Waypoint_1_p, qnear=Waypoint_1_q),
a=1.3962634015954636, v=1.0471975511965976)
    else:
        $ 5 "Else"
        $ 6 "MoveJ"
        $ 7 "Waypoint_2"
        movej(get_inverse_kin(Waypoint_2_p, qnear=Waypoint_2_q),
a=1.3962634015954636, v=1.0471975511965976)
    end
end
end

```

L'inconvenient d'aquest llenguatge és que desapareix la part intuïtiva i, per tant, es converteix en un hàndicap per l'operari que desconeix aquest llenguatge de programació però sabia utilitzar la programació a través del PolyScope.

El tercer mètode de programació consisteix en la combinació d'aquests dos mètodes anteriors, una combinació de la programació intuïtiva amb opció d'afegir part de programació directament amb codi URScript la qual cosa permet desenvolupar aplicacions més complexes.

En el cas d'aquest treball, el llenguatge utilitzat per la programació del robot la programació intuïtiva amb línies de codis entrats en format d'Scripts. Per la programació de les aplicacions que es gestionen des de l'ordinador s'utilitza el llenguatge de Matlab.

### 3.2.5. Entorn de programació extern i simulació:

Tal com s'ha comentat en l'apartat anterior, la programació es pot dur a terme en la totalitat en el PolyScope. Això permet no haver de dependre d'un ordinador extern. Tot i que això soluciona un problema, també és interessant disposar d'una alternativa de programació per poder desenvolupar els codis sense haver de tenir el robot. La proposta del fabricant UR és posar a disposició una màquina virtual que simula amb exactitud la tauleta de programació. Quan s'executa la màquina virtual que s'anomena URSim (Universal Robot Simulator), permet escollir el model del robot que es vol

programar. Una vegada oberta l'aplicació específica es podrà comprovar que és el mateix entorn de programació que el que disposa el PolyScope.

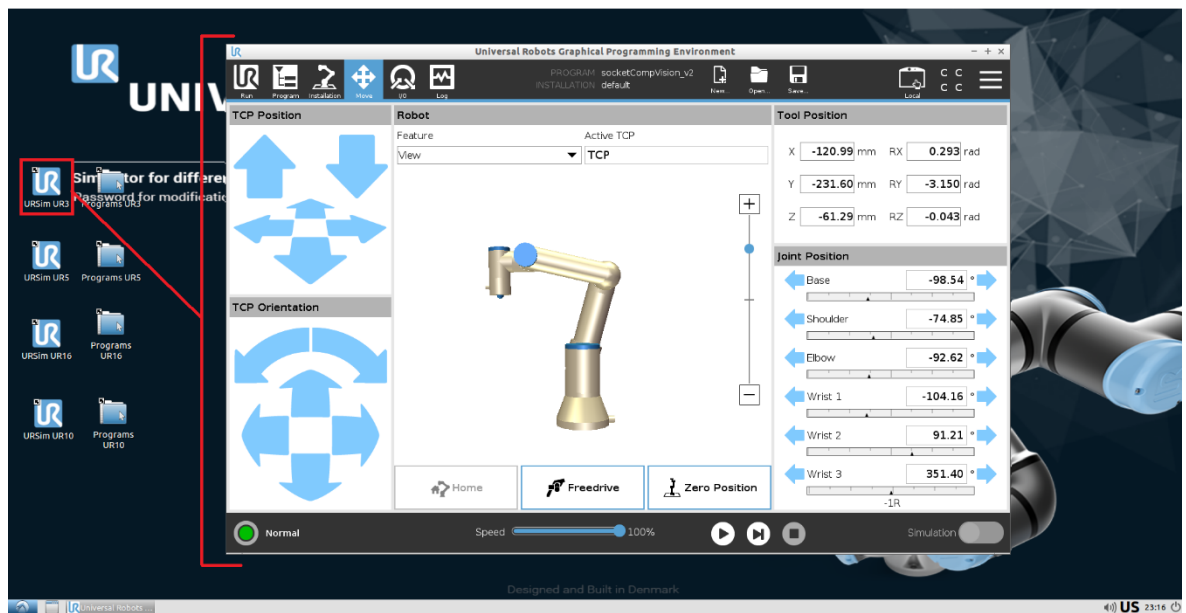


Figura 3.12. Captura màquina virtual i simulador URSim (Font: Bisnu Masó)

Una de les opcions més útil que incorpora és l'opció d'habilitar el mode de simulació, indiferentment des de quins dispositius s'utilitzi. A l'activar aquest mode, en la finestra de I/O, s'habiliten les entrades tipus *Digital Input*, *Tool Digital Input* i *Configurable Input* per poder forçar el seu estat i veure si el programa funciona tal com s'ha desitjat. S'ha de tenir en compte que en el mode de simulació es poden accedir en totes les finestres excepte la de moure el robot. Si es prem aquest botó, es desactiva el mode de simulació.

Aquesta funció és una eina de gran utilitat en l'àmbit educatiu i l'àmbit professional donat que permet crear programes sense necessitat de tenir tot el conjunt dels dispositius d'entrada, simular el seu funcionament i instal·lar-lo sense necessitat de parar la producció o la línia que conté el robot.

El mode de simulació s'activa amb el botó de la part inferior dreta de la pantalla i una vegada està activat es posa la finestra amb color blau per indicar aquest estat. Quan aquest mode està habilitat en el PolyScope del robot, el robot passa de l'estat *Normal* al *Simulation* i el robot físic queda desactivat. En el cas d'utilitzar el URSim, succeeix el mateix però al no tenir cap robot físic no cal tenir en compte aquesta part.

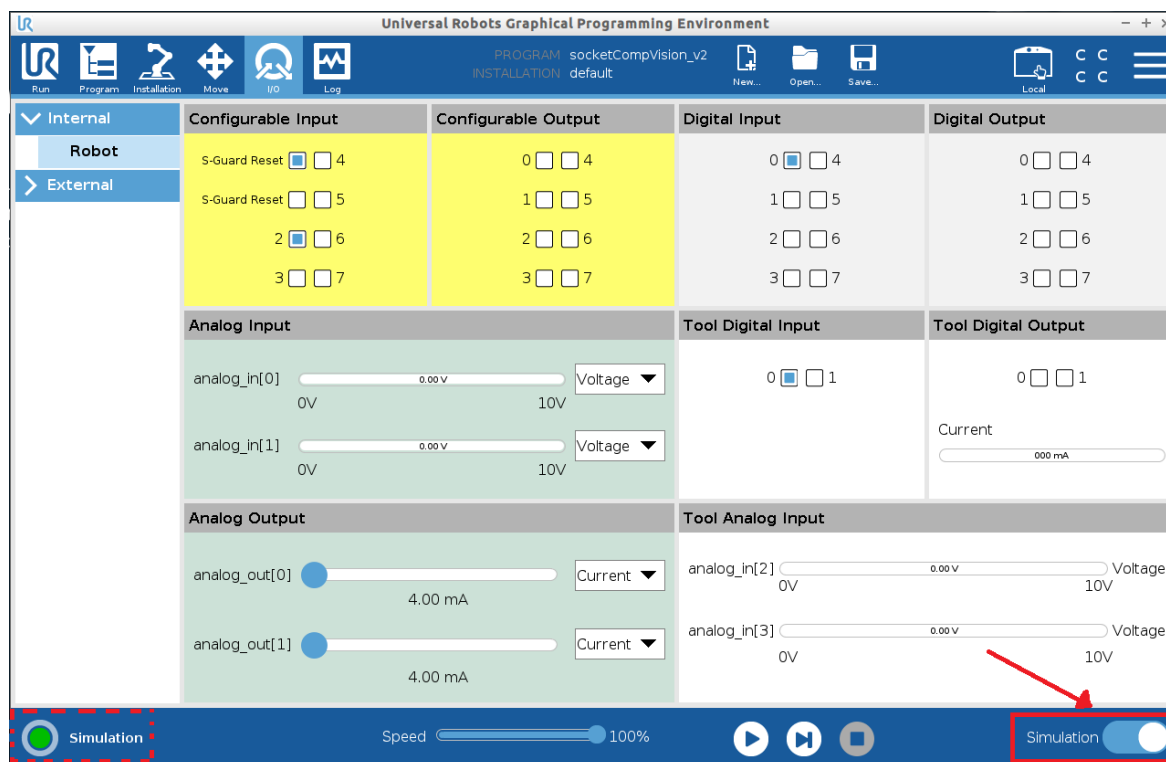


Figura 3.13. Captura pantalla I/O i mode simulació activat (Font: Bisnu Masó)

## 4. Muntatge robot

En aquesta secció es detalla el procés i les pautes del muntatge del robot i la seva col·locació en segons les diferents distribucions disponibles en el laboratori de robòtica del centre A5.4.

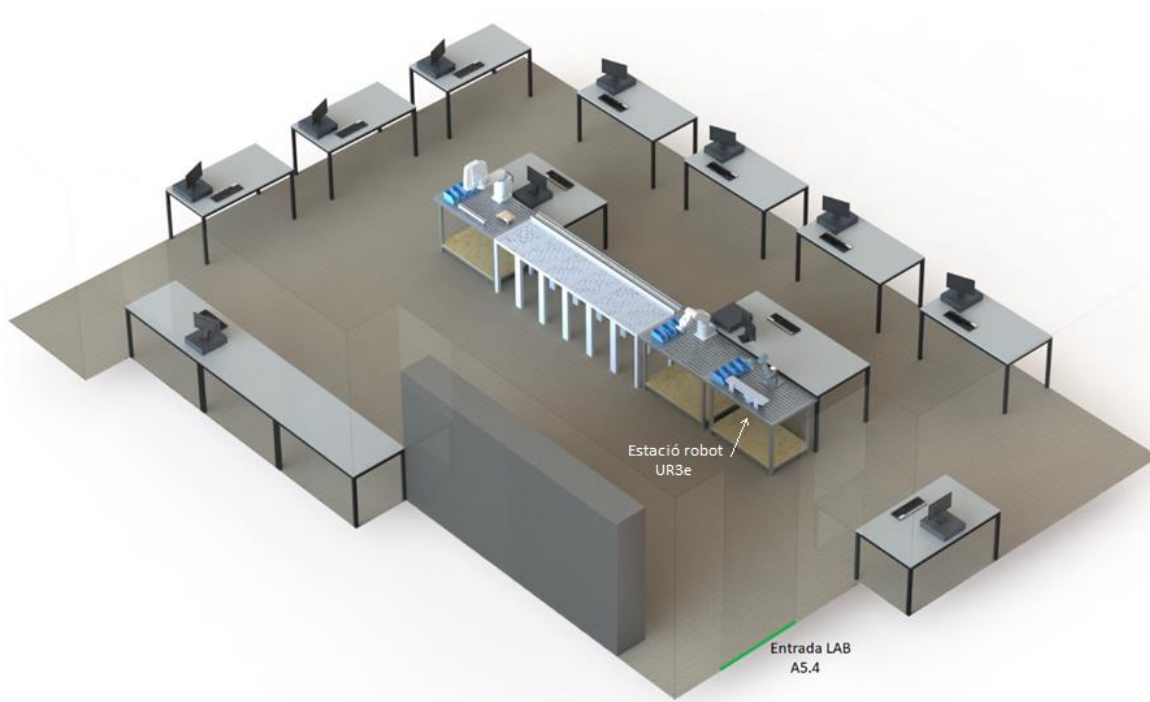
### 4.1. Disposició (*layout*)

Quan es fa una nova incorporació de maquinària, tant si és a nivell educatiu o a nivell industrial, realitzar un estudi per determinar el lloc d'instal·lació. Quan es tracta d'una incorporació en una indústria el lloc està més definit donat que la compra s'ha efectuat pensant en una tasca concreta i el lloc ja està definit per l'organització del local. En canvi, a l'incorporar aquest mateix equip en un centre docent cal valorar quines necessitats ha de satisfer i d'aquesta manera escollir la seva col·locació.

En el cas de l'estació del robot UR3e, es varen tenir en compte els següents aspectes:

- Ha de poder ser una estació mòbil per si s'ha de treure fora de l'aula.
- Ha de permetre un accés fàcil en els controladors.
- Espai ampli al seu voltant per permetre una manipulació sense obstacles.
- Accés a la xarxa d'alimentació i a la xarxa local Ethernet.
- Moure el mínim, a la mesura del possible, la resta d'equipaments.

El lloc on satisfà tots aquests aspectes és al costat de l'estació del robot articulat Mitsubishi. D'aquesta manera s'obté una línia central robotitzada permetent un accés des de qualsevol lloc de treball dels estudiants. A més a més té un accés fàcil al sistema neumàtic en cas que en un futur es vulgui utilitzar; accés a la xarxa local o xarxa externa per un control remot del robot, disposa d'un ordinador proper per si es volen dur a terme connexions entre el robot i el PC per tasques de, per exemple, visió per computador; permet accedir al robot per la part frontal i lateral dret; i no s'ha de moure ni modificar cap dels sistemes que hi ha actualment. En la següent figura es pot veure la seva localització respecte a la disposició de l'aula.

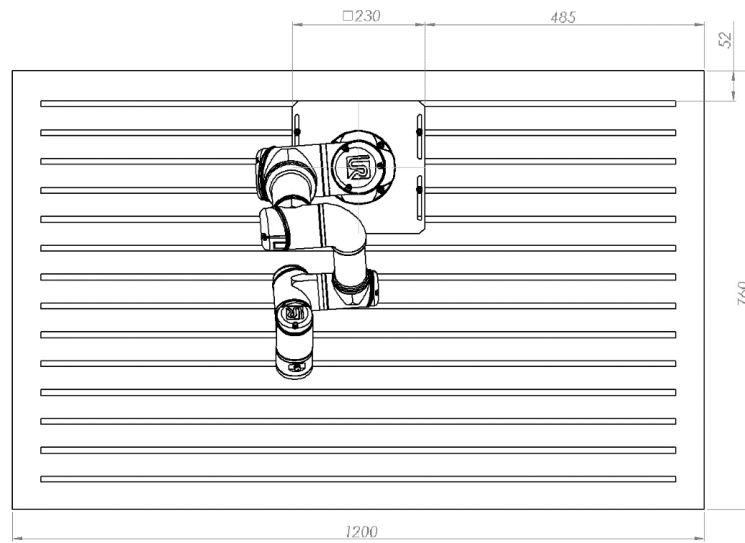


**Figura 4.1.** Distribució i *Layout* aula A5.4 amb la nova estació UR3e (Font: Bisnu Masó)

## 4.2. Instal·lació

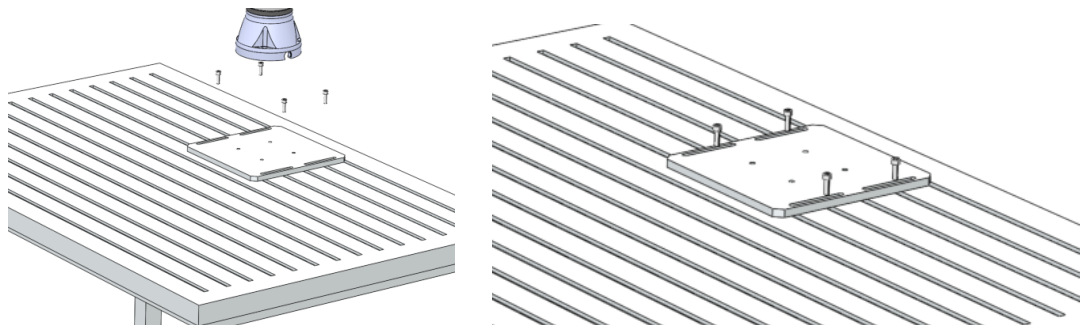
Una vegada decidit el *layout* es va seguir amb el muntatge de l'estació de robot en una taula d'alumini de la marca SMC. El conjunt està format pel mateix robot, la seva caixa de control i la platina de subjecció. El fet d'incloure tots els components en una taula amb rodes té com a objectiu facilitar el seu desplaçament per si s'ha de portar en alguna exposició o per poder-lo utilitzar en els possibles tallers, cursos, demostracions... que es puguin efectuar.

La disposició del robot en la taula és tal com es mostra en la següent figura de tal manera que la major part lliure de la taula està al seu abast, és a dir, en el seu espai de treball.



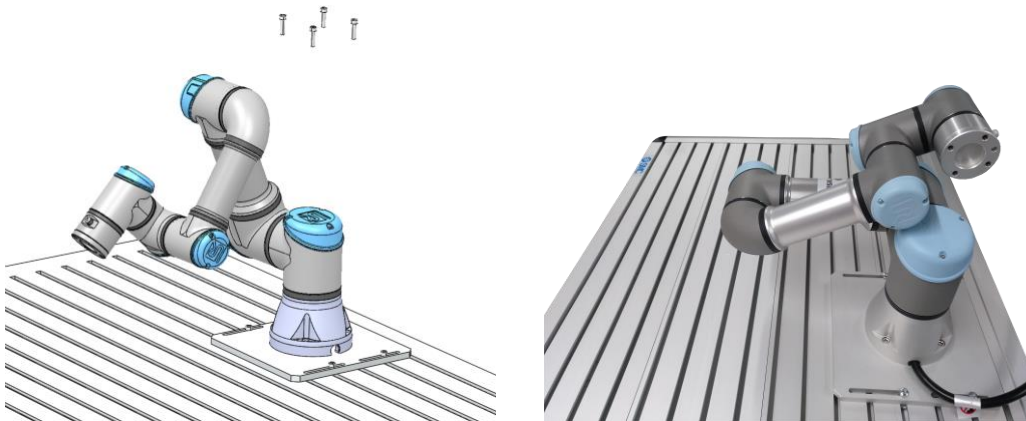
**Figura 4.2.** Distribució robot en la taula de treball (Font: Bisnu Masó)

El primer pas és el muntatge de la platina d'alumini de 23 cm de costats amb quatre cargols M8x15 amb volanderes i les femelles corresponents pel tipus de taula. La col·locació és la que es mostra en la figura anterior, centrada en la longitud de la taula i a 52 cm de la part superior. L'objectiu de la platina és obtenir una major subjecció entre la taula i el robot i una major estabilitat.



**Figura 4.3.** Col·locació i muntatge platina suport robot (Font: Bisnu Masó)

El segon pas és treure el robot de la caixa i muntar-lo en la taula. Per això seran necessaris quatre cargols M6x30 i quatre volanderes. El robot es centra en la platina i no es necessiten femelles per la seva fixació donat que els orificis que incorpora ja estan roscats. L'orientació del robot està col·locat de tal manera que mirat frontalment, el cable que surt del robot es situa a la part dreta en paral·lel a les cavitats de la taula. Es va triar aquesta posició per facilitar la seva manipulació quan parteix de la posició de repòs.



**Figura 4.4.** Muntatge i fixació robot amb la platina (Font: Bisnu Masó)

L'últim pas és col·locar la caixa del controlador i connectar-hi el cable del robot i el d'alimentació donat que la consola de programació no es pot desconectar de la caixa. Els dos connectors estan situats en la part inferior de la caixa en la part dreta. El connector d'alimentació és l'estàndard IEC 60320 C20, comú entre els dispositius domèstics amb un voltatge màxim d'alimentació de 250 VAC i es connecta a pressió. El connector del cable del robot és un adaptador circular on s'hi han de fer coincidir els pins de connexió, empènyer el connector i fixar-lo amb un moviment de rotació de 180º



**Figura 4.5.** Caixa controlador robot UR3e (Font: Bisnu Masó)

Tal com es pot veure, a diferència d'altres marques i models de robots, el muntatge és senzill i no s'hi aprecien dificultats amb nocions bàsiques d'electrònica.



## 5. Disseny i construcció de complements

En aquest apartat es detallen les diferents peces dissenyades i construïdes: un element terminal amb un conjunt de peces per treballar-hi, una botonera que serveix com a emulador de sensors digitals, muntatge d'una caixa de connexions i un suport per posar-hi la consola de programació (també anomenat *teach pendant*).

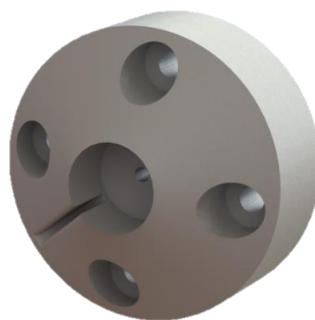
### 5.1. Element terminal

#### 5.1.1. Versió 1

L'element terminal és el sistema del robot que permet interaccionar amb el seu entorn. Donat que és un aparell d'ús industrial on es sol dissenyar un element terminal per una aplicació en concret, el robot no ve amb cap element terminal de fàbrica. Com que els preus per uns manipuladors són elevats i les gestions per comprar-los solen ser de termini llarg, es va decidir construir un element terminal senzill que permetés poder tenir el robot operatiu i funcional en pocs dies.

Dels dos sistemes d'accionament disponibles, pneumàtic i elèctric, es va descartar el pneumàtic per no dependre d'un compressor si l'estació ha de poder ser mòbil. Dels diferents tipus que hi ha amb sistemes elèctrics, es va acabar decantant per un electroimant al ser un actuator molt senzill envers el sistema de pinces.

Una vegada obtingut l'electroimant es va procedir en dissenyar l'element que permetés subjectar-lo a l'extrem del robot. Un cop dissenyat es va imprimir amb una impressora 3D. El material de fabricació és amb PLA (àcid polilàctic), un termoplàstic amb material base provinent de fonts vegetals. El disseny final es mostra en la figura 5.2.

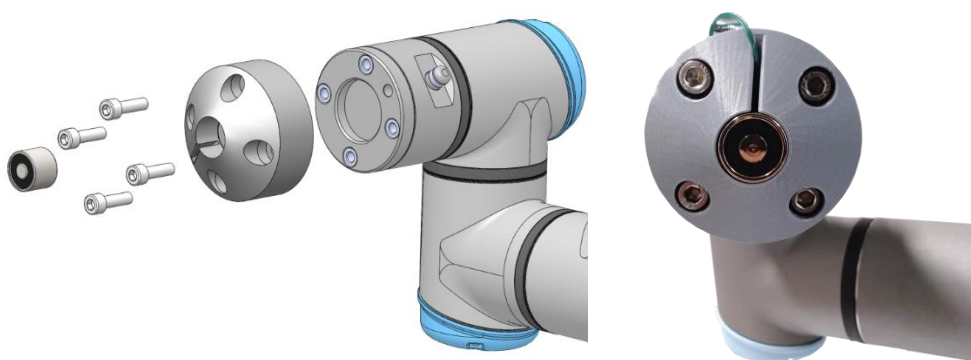
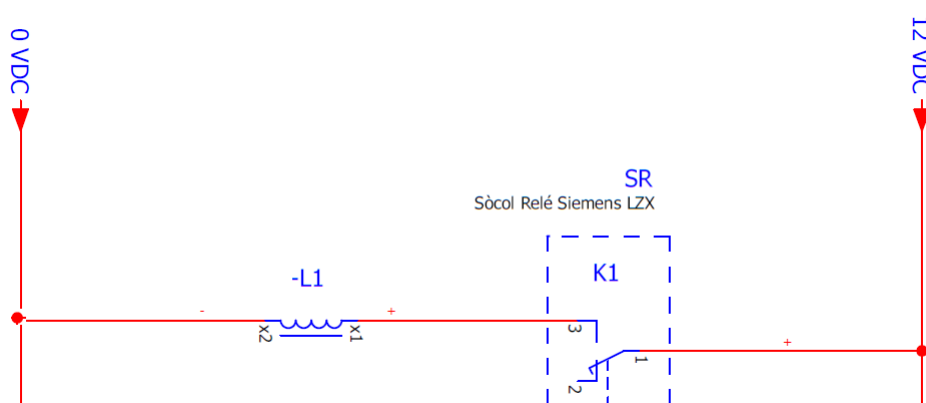


**Figura 5.1.** Disseny element terminal per electroimant (Font: Bisnu Masó)

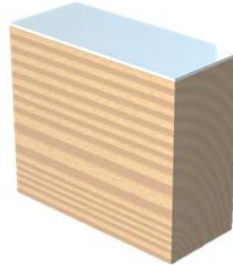
**Taula 5.1.** Especificacions electroimant XRN-XP.

Especificacions electroimant XRN-XP		
Paràmetre	Valor	Unitat
<b>Tipus</b>	Solenoides	-
<b>Voltatge nominal</b>	12	VDC
<b>Amperatge</b>	0,24	A
<b>Atracció max</b>	2,5	kg
<b>Dimensions (<math>\varnothing</math> x Altura)</b>	20x15	mm
<b>Pes</b>	25	g

Finalment, després de comprovar que totes les peces encaixessin, es va procedir a muntar-ho en el robot. En la següent imatge de l'esquerra es pot veure les diferents parts de l'element terminal format per l'electroimant de 12 VDC, els cargols M6x20 i l'adaptador. En la imatge de la dreta, l'element terminal acoblat al robot real.

**Figura 5.2.** Instal·lació element terminal i visió final (Font: Bisnu Masó)**Figura 5.3.** Esquema elèctric connexió electroimant (Font: Bisnu Masó)

Paral·lelament, com que l'element dissenyat només pot ser utilitzat amb elements fèrrics al basar-se amb un electroimant, es varen construir unes peces de fusta amb una planxa metàl·lica en la cara superior i així obtenir un robot funcional en menys de dues setmanes.



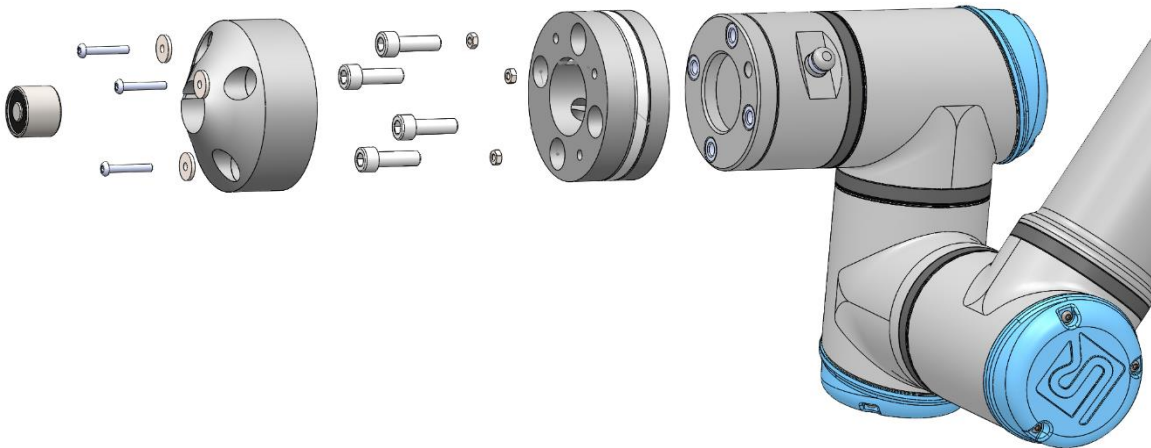
**Figura 5.4.** Peça de fusta amb planxa metàl·lica (Font: Bisnu Masó)

### 5.1.2. Versió 2

En l'anterior versió de l'element terminal no era possible saber l'estat operatiu de l'actuador donat que és un electroimant i no té mobilitat.

En un àmbit industrial, com la majoria de sensors, incorporen uns petits LEDs que indiquen l'estat. Com que l'ús d'aquest robot és en àmbit acadèmic, és convenient incorporar un sistema que permeti veure l'estat operatiu de l'element terminal de la manera més visible possible. Per satisfer aquesta necessitat s'ha optat per dissenyar una peça que serveixi com a adaptador que incorpora una tira de llums LEDs RGB que canvia de color segons si l'electroimant està activat o desactivat. Aquesta peça, igual que el suport per l'actuador, s'ha construït i imprès amb una impressora 3D a partir d'un disseny propi i millorat que permet passar els fils elèctrics per l'interior de la peça a través del forat que incorpora en un lateral. L'avantatge que presenta aquesta peça és que pot ser fixe i canviar només el bloc final podent-hi acoblar altres tipus d'elements terminals com per exemple: elements de dibuix, fresadora, diferents tipus d'eines, etc.

El procés de muntatge es pot veure en la següent figura. La peça que incorpora les llums es fixen amb quatre cargols M6X20 que permeten una bona fixació. L'electroimant es fixa al seu suport amb un cargol M3X20 per al part posterior d'aquest suport (no visible en la figura). Finalment, el conjunt final d'element terminal, es fixa amb la peça que conté les llums amb tres cargols M3X30 que es cargolen en les femelles que s'incorporen en la primera peça. Per millorar el muntatge, s'utilitzen 3 volanderes amb diàmetre exterior adequades al diàmetre de l'orifici i amb diàmetre interior de 3,5 mm. Finalment s'hi ha afegit un fil extensible i s'han efectuat les connexions pertinents tal com es mostra en la figura 5.6.



**Figura 5.5.** Versió final de l'element terminal amb adaptador d'ús genèric. (Font: Bisnu Masó)

La relació de colors escollida per aquest actuador són els següents:

- Electroimant activat -> color **VERD** (LEDs color verd encesos).
- Electroimant desactivat -> color **MORAT** (LEDs color blau i vermell encesos)



**Figura 5.6.** Element terminal amb indicador lluminós en funcionament (Font: Bisnu Masó)

## 5.2. Simulador sensors digitals

Un simulador és un dispositiu que permet representar el funcionament d'un element real. Quan no es té a disposició aquest element o per fases de proves, és una eina molt útil a l'hora de desenvolupar i provar el programa. Per això, es va dissenyar una botonera que, en la primera versió, incorporava sis interruptors i un polsador per simular diferents sensors digitals. En cas de voler simular sensors analògics s'hi podria incorporar potenciómetres i treballar amb els diferents valors de voltatge que rep l'entrada analògica. La versió final és la que es troba en la figura 11.4 de l'apartat 11.2 que es va optimitzar per deixar entrades digitals lliures per la incorporació de sensor reals.

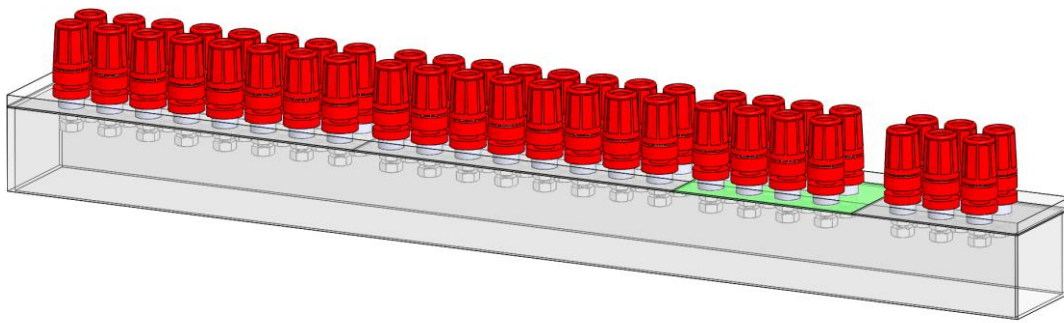


**Figura 5.7.** Versió inicial simulador sensors digitals (Font: Bisnu Masó)

### 5.3. Muntatge sistema elèctric de connexions

En el mes i mig d'experiència i contacte que es va tenir amb el robot, es va poder comprovar que el sistema de connexions amb els terminals del robot no són un sistema còmode per connectar i desconectar els cables al ser d'unes dimensions reduïdes en un espai bastant tancat. Alhora, el fet d'estar connectats entre ells de forma compacta és costós treure els terminals de la caixa amb facilitat i no permet fer connexions sense treure l'adaptador donat que el cargol queda tapat amb els adaptadors dels costats. Per resoldre aquesta problemàtica, s'ha optat per dissenyar una caixa amb connectors del tipus banana, com els que s'utilitzen en els laboratoris d'electrònica. En aquests connectors femella s'hi porten les connexions de la caixa del controlador a la nova placa de connexions que es situaria a la part superior de la taula del robot. D'aquesta manera s'evita el desgast de treure i posar els terminals, es protegeixen la resta de connexions que conté el controlador i resulta còmode a l'hora de posar i treure components segons el tipus de pràctica que es vulgui realitzar.

El disseny final de la placa de connexions té una forma allargada per reduir el màxim l'espai que ocupa i així permetre un espai de treball major pel robot. Respecte a la idea inicial on s'utilitzava una caixa per encastar de la marca Schneider Electrònics, al no tenir-la disponible a casa, s'ha utilitzat una placa de metacrilat amb diferents orificis on s'hi col·loquen els connectors banana femella. Per diferenciar clarament els diferents ports, s'ha col·locat un cartró del mateix color que els terminals de la caixa amb les marques corresponents a cada connector. El resultat és el que es mostra en la següent figura:

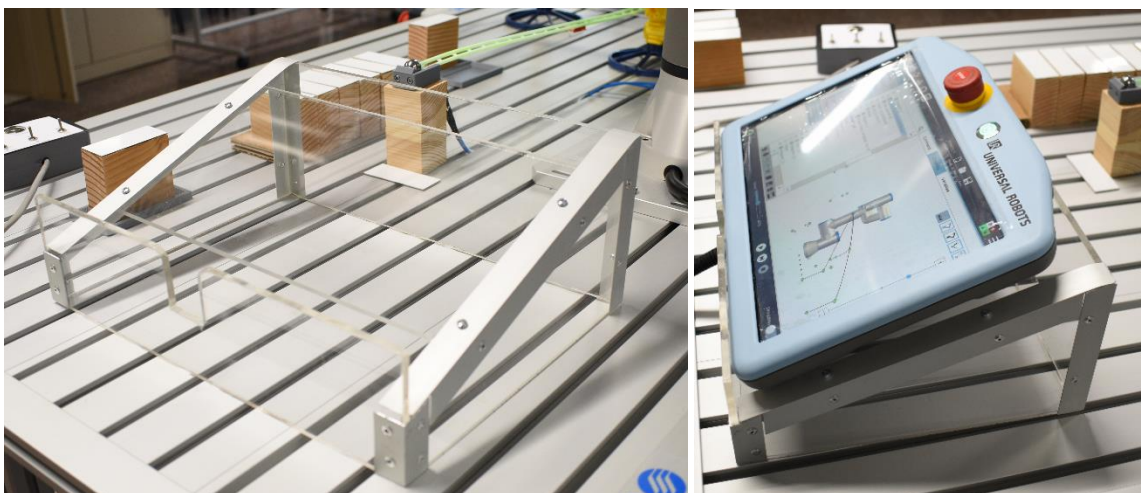


**Figura 5.8.** Caixa de connexions banana d'I/O del controlador (Font: Bisnu Masó)

## 5.4. Construcció suport consola de programació

A diferència d'altres robots i consoles de programació, la del robot d'Universal Robots és una pantalla tàctil que permet que sigui més interactiva i visual i permeten mostrar elements que altres consoles de robots més tradicionals no ho permeten. Alhora també permeten la introducció de dades o sol·licitar la intervenció de l'usuari a partir de POP-UPS i altres tipus de missatges. Per aquest motiu, és necessari tenir el *teach pendant* accessible. Per facilitar l'accés i la correcta visualització s'ha construït un suport que permet asseure-hi el dispositiu per una millor manipulació i angle de visió quan no es té a les mans.

El material escollit per la construcció de l'estructura principal és el metacrilat de 5 mm de gruix. Per unir les diferents parts entre si i obtenir unes unions fermes, s'han utilitzat perfils d'alumini en forma de L de 20 mm de costat i s'han utilitzat reblons cecs per fixar els perfils amb el metacrilat. El fet d'utilitzar reblons i no cargols és la facilitat de muntatge i els acabats més senzills que permeten obtenir.



**Figura 5.9.** Disseny i construcció suport consola de programació (Font: Bisnu Masó)

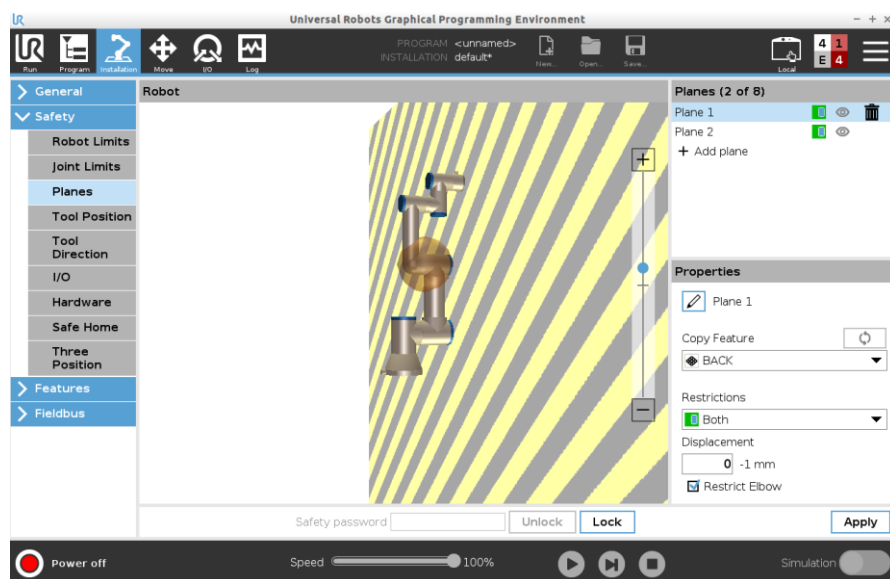
## 6. Posada en marxa

Una vegada el robot està instal·lat al seu lloc i ja s'hi ha col·locat l'element terminal és important la primera configuració. Per una banda per tenir un sistema més segur i per altra, poder controlar el robot amb exactitud i moviments ajustats amb el TCP (Tool Center Point).

### 6.1. Seguretat

El primer pas més important a dur a terme és augmentar la seguretat de l'estació. En aquest aspecte inclou la instal·lació d'un botó de parada d'emergència i la definició de plans de seguretat (parets virtuals) que el robot no traspasarà en cap circumstància. D'aquesta manera es defineix l'àrea de treball del robot i permet evitar col·lisions amb els elements del seu entorn.

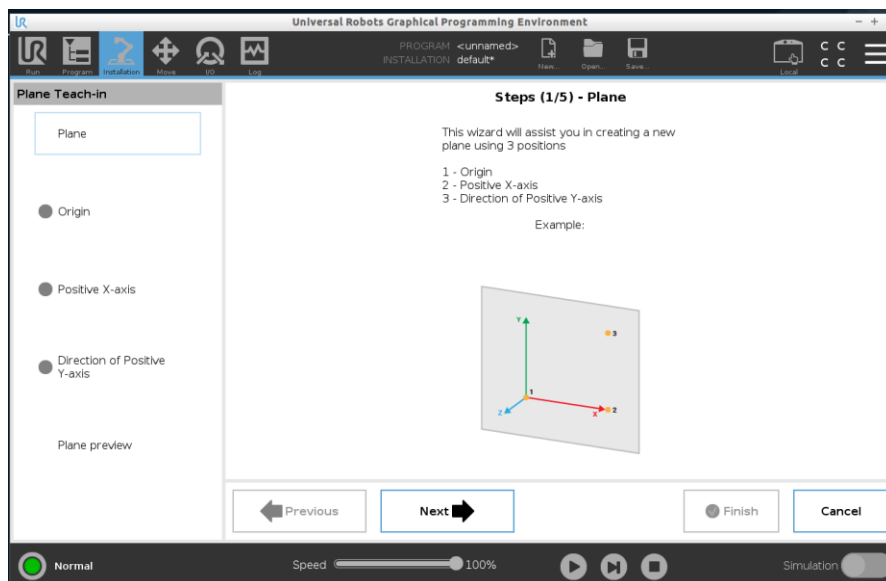
Per accedir i modificar configuracions de seguretat primer de tot s'ha d'establir una contrasenya al robot per aquests paràmetres. La contrasenya s'estableix a: *Menu > Settings > Password > Safety*. Tot seguit, a la finestra *Installation > Safety > Planes* es poden definir fins a 8 plans de seguretat.



**Figura 6.1.** Captura definició pla de seguretat vertical (Font: Bisnu Masó)

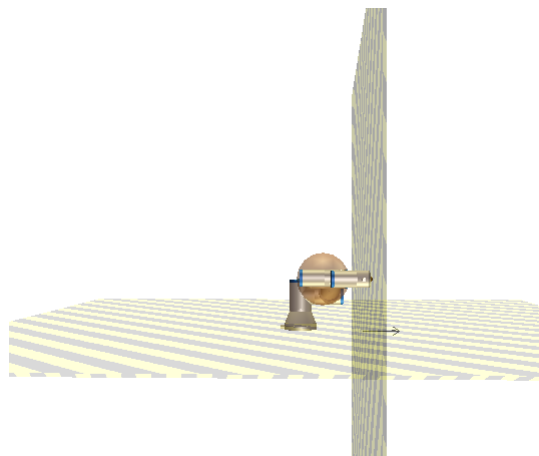
Per defecte només hi apareix un pla respecte a la base, per afegir altres plans cal crear-los a *Installation > Features > Plane* i seguir les instruccions que apareixen per pantalla:





**Figura 6.2.** Captura com afegir nous plans (Font: Bisnu Masó)

En el cas del robot del laboratori s'han definit 2 plans, un corresponent a la superfície de la taula per evitar col·lisions, amb una distància de seguretat de 5 cm per evitar pinçaments, i l'altra en la part posterior per evitar que sobresurti de la taula i pugui col·lidir amb l'equipament de la taula de l'ordinador.



**Figura 6.3.** Plans de seguretats definits en el robot UR3e (Font: Bisnu Masó)

Una vegada s'ha establert els plans de protecció i es guarda la configuració, el símbol de suma de comprovació de seguretat canvia a l'haver afegit plans de limitació i per tant, haver canviat les funcions de seguretat.





Figura 6.4. Canvi icona suma de comprovació al afegir plans de seguretat (Font: Bisnu Masó)

## 6.2. Determinació del nou TCP

En primer lloc cal diferenciar el concepte de TCP d'aquest apartat al qual fa referència a les sigles de l'anglès *Terminal Center Point* i no del TCP de l'apartat de comunicacions que feia referència al *Transmission Control Protocol*.

El TCP, punt central de l'element terminal, es defineix com la diferencia de translació i rotació respecte a la brida del robot (punt on s'uneix l'element terminal amb el robot) i la punta de l'element terminal. Tots els moviments i posicions del robot vindran determinats i referenciats per la localització i posicionament del TCP. En el següent exemple de la figura es pot veure la importància de definir un TCP. En els dos casos la posició inicial i la final és la mateixa però la trajectòria que efectua el robot és diferent. En la figura de l'esquerra, on no s'hi ha definit cap TCP, es pot veure que qui fa el moviment lineal és l'articulació i no la punta de l'eina que aquest incorpora. En la figura de la dreta, on s'ha definit un TCP situat a la punta de l'element terminal, es pot veure que qui fa el moviment lineal és el TCP i, en canvi, el robot fa el moviment que sigui necessari perquè es pugui dur a terme aquest desplaçament lineal. En el primer cas és imprevisible determinar quin serà el recorregut i les posicions de l'element terminal, en canvi en el segon cas, el que mana és el TCP i la seva posició o trajectòria podrà ser coneguda.

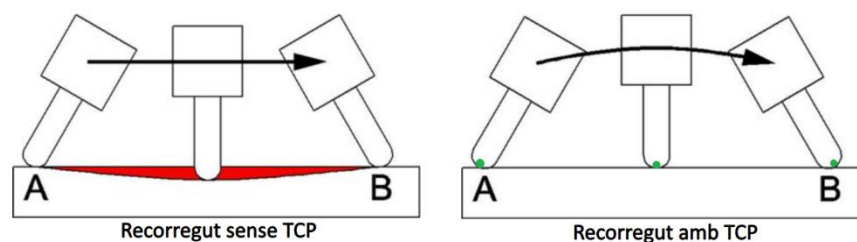
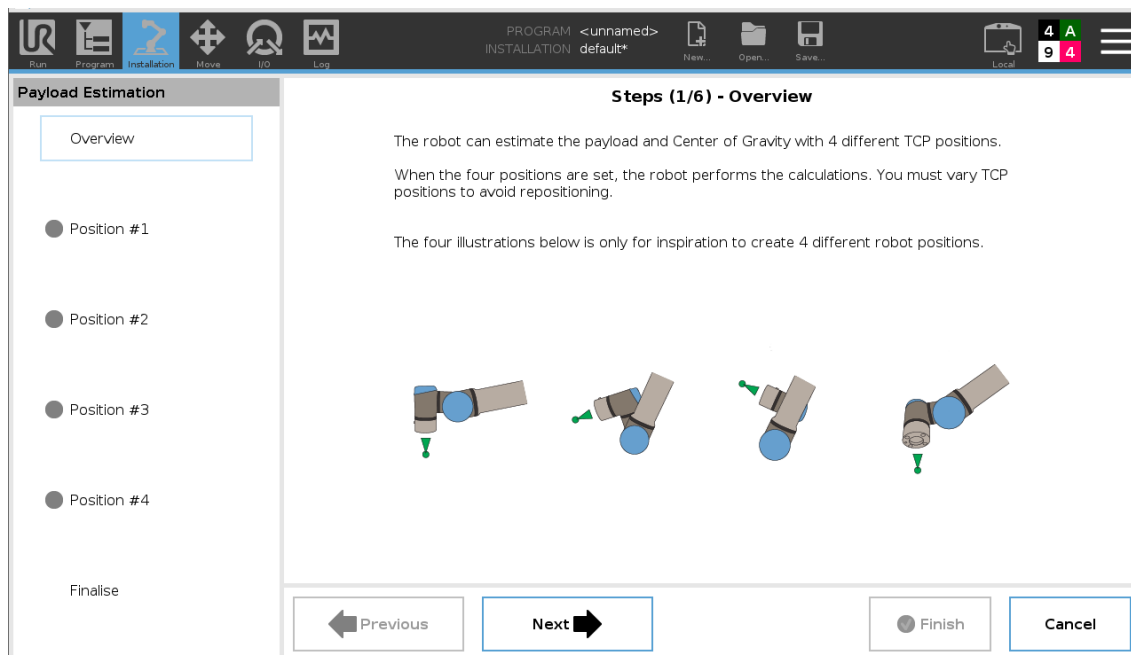


Figura 6.5. Diferència de recorregut amb i sense TCP definit (Font: [34])

En el cas del robot de laboratori, una vegada es va col·locar l'element terminal, abans de començar amb la programació es va començar per configurar el TCP. Per configurar-lo s'ha d'anar a la pestanya *Installation > General > TCP*. Tal com es mostra, accedint al botó *Wizard* permet configurar la posició, l'orientació o la càrrega (pes element terminal) i el seu punt de gravetat. Si s'accedeix directament a

l'última opció i es segueixen els passos que apareixen en pantalla, al mesurar la carrega i el centre de gravetat, automàticament es calcula l'orientació i la posició del TCP.



**Figura 6.6.** Captura definició TCP i altes paràmetres de l'element terminal (Font: Bisnu Masó)

Una vegada configurat aquests paràmetres, es va procedir en fer proves de funcionament de l'equip.

## 7. Estudi de la força

### 7.1. Transductors de força

Un transductor és un dispositiu format per cèl·lules sensibles que detecten magnituds físiques o químiques i les converteixen en senyals elèctrics que permeten tractar i estudiar la informació més fàcilment. Per tal de mesurar la força es mesura l'efecte que provoca la força en un element el qual varia la seva dimensió, genera càrregues o canvia alguna de les seves propietats. Per detectar la força es poden utilitzar diferents tipus de transductors que utilitzen principis diferents, els més comuns són els següents:

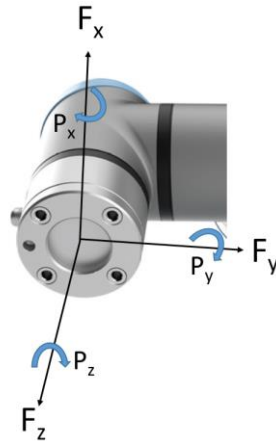
- **Galgues extensomètriques:** es basen en el canvi de la resistència elèctrica pel pas d'un material quan s'aplica una força o tensió mecànica. Aquesta variació es produeix per les deformacions longitudinals causades en el material [35].
- **Piezoelèctrics:** es basen en generar diferència de potencial en aplicar-hi una força. El material més comú utilitzat per aquest tipus de sensor és són els cristalls de quarts que generen una càrrega electroestàtica proporcional a la força aplicada.
- **Capacitiu:** format per un material on la seva capacítancia varia quan s'aplica una força o tensió mecànica. A partir de l'electrònica que si afegeix permet obtenir una tensió analògica a partir d'estímul externs.
- **Reluctàncies:** es basa en la variació de la resistència en el pas del flux magnètic quan s'incideix un camp magnètic en un material conductor. A partir de la variació del flux d'electrons dins del material, permet calcular la força aplicada a l'apropar el conductor al camp magnètic.

Aquest tipus de dispositius tenen camps d'aplicacions molt amplis: en l'enginyeria per estudis de deformacions i esforços en estructures, en la medicina per la mesura de pressió i el treball i forces en les articulacions, en l'agronomia per mesurar la força en els estris de llaurar, en la construcció de bàscules, etc. En l'àmbit de la indústria i la robòtica els sensors de força s'utilitzaven principalment en aplicacions com la manipulació d'objectes o operacions de mecanitzat. En l'actualitat aquests sensors s'incorporen en els robots per tal de limitació de potència i força i obtenir així un robot col·laboratiu.

### 7.2. UR3e i la força

Tal com s'ha comentat en l'apartat de seguretat, el robot UR3e incorpora un sistema avançat de control de força en cada un dels eixos que permet ajustar-ho entre 100 i 250 newtons. D'aquesta manera, al poder limitar la força i la potència, permet al robot treballar amb entorns sense necessitat de barreres de seguretat addicionals fent-lo ideal per entorns reduïts.

Per altra banda, a part dels sensors en cada eix, incorpora un sensor de força capaç de detectar força i parell en el sistema de coordenades de 3 eixos (X, Y, Z). Aquest sensor és més precís i es troba situat en l'anella on s'hi col·loquen els elements terminals i permet obtenir uns resultats amb una precisió de 0.1 Nm amb una càrrega màxima de 30 N.



**Figura 7.1.** Eixos de detecció força i parell del sensor de força (Font: Bisnu Masó)

A través d'aquest sensor es poden dur a terme aplicacions de precisió on sigui necessari tenir en compte la força aplicada. Una aplicació de les més senzilles que incorpora en forma de funció avançada és la de cargolar i descargolar cargols (Screwdriving). Per utilitzar aquesta funció primer de tot s'han de determinar diferents paràmetres en la pantalla *Installation > General > Screwdriving*: seleccionar el TCP, el senyal d'encesa del motor, selecció del tipus de programa i els senyals de notificació. Una vegada definits els paràmetres, en la pantalla de programació, utilitzant la funció esmentada anteriorment, permet configurar el procés. El primer pas és determinar si es vol cargolar o descargolar, el segons pas és determinar el tipus de procés: força /parell que s'aplica i la velocitat límit i, finalment, el tercer i últim pas, determinar quin és el senyal de finalització del procés: distancia, temps, expressió...

\* Nota: el codi de la següent captura no està complet. Falten els passos de posicionament inicial i final.

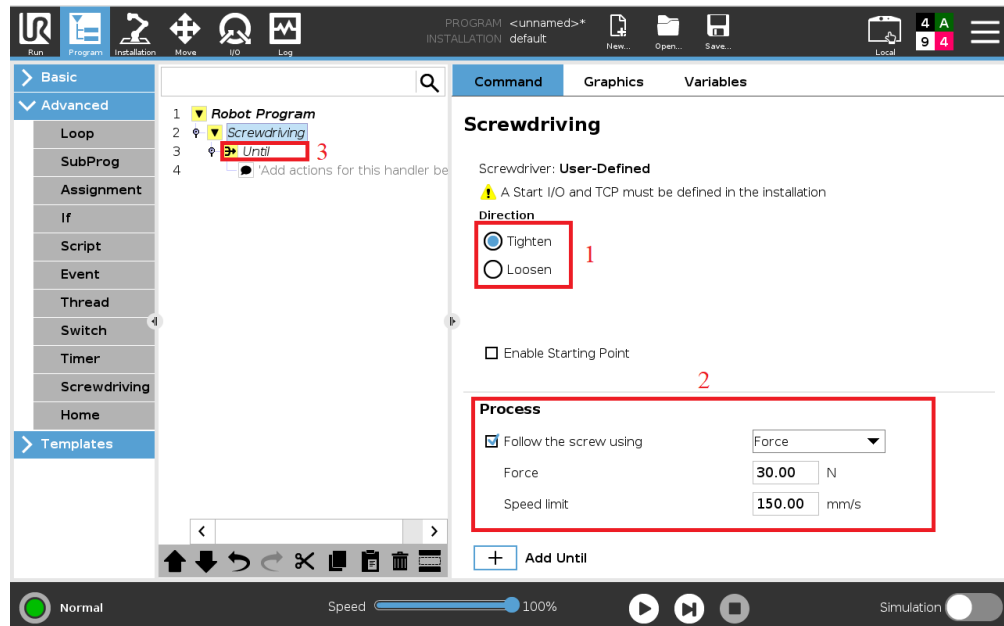


Figura 7.2. Captura funció Screwdriving i paràmetres a ajusta (Font: Bisnu Masó)

Per altra banda, per obtenir més funcionalitats es pot utilitzar la plantilla *Force* que es troba a *Program* > *Templates* > *Force* on està format per quatre modes:

- **Simple:** el robot només aplica la força en un sentit, eix Z, segons la funció seleccionada. Segons si és la base o el TCP el robot es mourà en la seva direcció Z aplicant la força especificada.

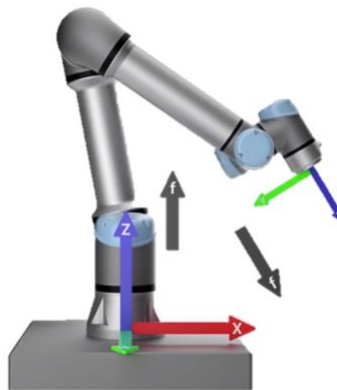
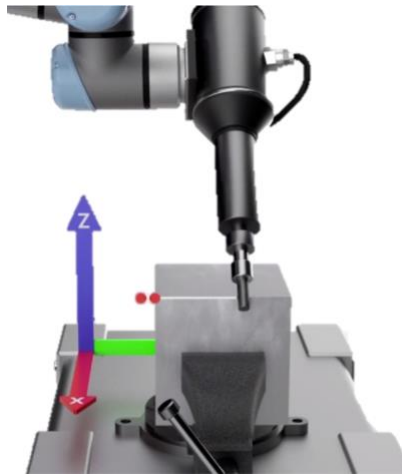


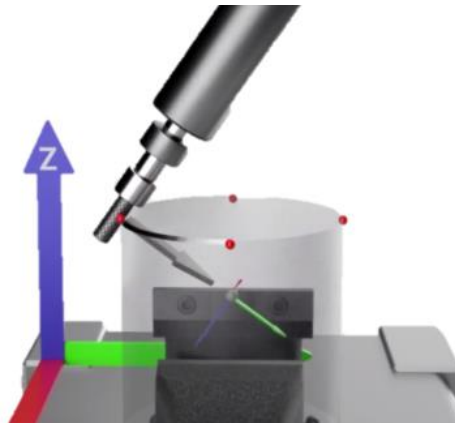
Figura 7.3. Eixos i sentit de la força del mode simple (Font: Bisnu Masó)

- **Frame:** el robot aplica força en més d'una direcció a partir d'una funció de pla de treball definit. Permet definir la força en cada un dels eixos i el parell a aplicar si és necessari.



**Figura 7.4.** Exemple d'us del mode de força *frame* (Font: Bisnu Masó)

- **Point:** el robot aplica força en l'eix indicat la força de la qual convergeix cap al punt definit.



**Figura 7.5.** Exemple d'us del mode de força *point* (Font: Bisnu Masó)

- **Motion:** permet combinar qualsevol dels modes anteriors aplicant un desplaçament del robot donat que en els casos anteriors, normalment, els moviments del robot eren limitats i sense desplaçament gran.

Fins ara, totes les aplicacions que s'han explicat l'usuari és qui defineix la força que fa el robot sobre un element però també és possible utilitzar el robot per mesurar el pes d'una peça en un procés de *pick and place* o la força que l'usuari aplica en l'element terminal. Per obtenir aquesta mesura s'ha d'afegir una de les següents línies de codi en el programa a partir de la comanda *Assignment*.

```
force() 'Retorna els valors de força TCP (Fz, Fy, Fz) en Newtons (tipus variable: float)'  
  
get_tcp_force() 'Retorna els valors de força i parell en el TCP (Fz, Fy, Fz) i (Px, Py, Pz) en Newtons i Nm respectivament.'
```

```
1 ▼ Robot Program  
2 ■ ForcaParell:=get_tcp_force()  
3 ■ Forca:=force()
```

**Figura 7.6.** Línies de codi per la lectura de la força amb programació intuïtiva (Font: Bisnu Masó)

### 7.3. Aplicacions docents

En aquest apartat es proposen diferents aplicacions on s'utilitza el sensor de força.

- De la mateixa manera que una aplicació per visió per computador pot ser una bona solució per controls de qualitat, el control i detecció del pes també ho pot ser en processos de paletització de productes i aplicacions de *pick and place* per determinar si el producte conté la quantitat de “substància” correcta o ha de ser descartada com a defectuosa. En l'àmbit acadèmic es podrien dissenyar un conjunt de peces amb pesos diferents i crear un programa de classificació de peces segons el pes detectat.
- Donat que l'ús de màquines de fresat poden suposar un perill, una alternativa seria la de dissenyar un element terminal on s'hi pogués col·locar un estri de dibuix (llapis, bolígraf, guix, retoladors...) i elabora programes on el robot dibuixa figures tenint en compte la força aplicada per dibuixar.
- Ús del robot per estudis aproximats amb material i assajos mecànics de duresa, torsió, elasticitat... amb materials de propietats senzilles que no requereixin de grans maquinàries per dur-les a terme.

Dona que el simulador del robot no és possible aplicar-hi forces, no s'hi inclouen els codis de programa ni els resultats de les simulacions.

## 8. Estudi de la comunicació i accés remot

En aquesta secció s'estudia i es posa en pràctica els diferents mètodes de comunicació entre el robot i l'ordinador i d'aquesta manera poder ampliar les funcionalitats de l'estació. Per exemple: disposar d'un sistema de visió per computador on l'ordinador actua com a sistema de processat, control del robot des d'un software extern, anàlisis de variables de la màquina per estudis de rendibilitat, hores de treball o manteniment preventiu, etc.

Tots els codis són a base d'un mateix llenguatge de programació que s'utilitza en el software Matlab, un entorn de computació numèrica i programació. S'ha escollit el Matlab donat que és un programa que s'utilitza al llarg de la carrera universitària en aquesta universitat i té un conjunt de funcionalitats amb un gran potencial. Per tant, els codis que es presenten en els següents apartats s'executen amb el programa Matlab i es comuniquen amb el programa de simulació del robot URSim per no haver pogut accedir al robot degut al tancament del centre per la pandèmia del coronavirus.

En els primers apartats s'explica com dur a terme la comunicació i una aplicació útil de transferència de fitxers i en els següents apartats dues aplicacions de control del robot.

### 8.1. Accés remot fitxers

Per accedir als documents i fitxers dels programes que conté el robot es pot dur a terme de manera offline a través del port USB que integra la consola de programació o de forma remota a partir de l'accés a un servidor. El problema que presenta la connexió física és que no hi ha l'opció d'extreure el dispositiu amb seguretat quan es vols desconnectar el *pendrive* o qualsevol dispositiu USB. Això comporta que, a mig-llarg termini, el port pot malmetre's i deixar de detectar dispositius USB i la reparació pot comportar la substitució de la consola de programació sencera. Per altra banda, al ser una connexió de fàcil accés, està exposat a perills de seguretat informàtica: virus, còpia de programes confidencials, inserció de fitxers maliciosos, etc.

La segona opció permet transferir fitxers entre ordinadors de forma remota a través del protocol FTP (*File Transfer Protocol*) que es basa amb el model de connexió TCP/IP, un model de protocol utilitzat per tots els ordinadors connectats a internet. Per poder dur a terme una transferència de fitxer és necessari un client FTP, un ordinador que es connecta a un servidor FTP i té un accés total o limitat dels fitxers de l'ordinador que actua com a servidor. En el cas del robot d'Universal Robots, a través d'una aplicació client FTP com el FileZilla, WinSCP i Transmit, entre d'altres, es pot accedir als fitxers i carpetes del robot. El problema d'aquest accés és que tots els fitxers, inclosos els del sistema operatiu, són visibles i es poden modificar. Si un usuari, sense voler o de forma intencionada, modifica un fitxer del sistema, aquest pot fer-se malbé i deixar el robot inoperatiu.



Per solucionar el problema del desgast de la connexió USB i restringir l'accés als fitxers del sistema, s'ha creat un programa amb Matlab amb funcions bàsiques que permet pujar fitxers en una carpeta determinada del servidor FTP des de qualsevol ordinador que tingui aquest programa i estigui connectat al robot.

La següent aplicació s'ha creat pensat per donar accés als estudiants, a través d'un programa ja conegut i estudiat, el Matlab, en assignatures de: càlcul numèric i equacions diferencials, regulació automàtica, tècniques de control, robòtica industrial i visió per computador... La finalitat és que els estudiants puguin penjar el seu programa al robot i després poder-hi accedir a partir de la consola de programació sense la necessitat de connectar-hi un USB. D'aquesta manera es preserva el port per casos estrictament necessaris. Per altra banda, al ser un programa amb codi, encara que totes les comandes d'accés i modificació en un servidor FTP són públiques [36], és més difícil la modificació dels fitxers del sistema donat que s'han d'executar de forma implícita.

```
%% ----- Accés FTP al robot UR3e -----
% Per pujar documents des de l'ordinador al robot sense necessitat
% d'accedir a la FTP completa i editar/eliminar la resta de documents dels
% sistema que són crítics si es modifiquen.
% -----

%% https://www.mathworks.com/help/matlab/ref/ftp.html

%% Connexió amb el servidor FTP
ftpobj = ftp('ftp.onstatic-es.setupdns.net','ftp.ishower','rrhhis2019ftp')
dir(ftpobj) % Veure documents del directori

%% Obrir una subcarpeta per penjar els documents
cd(ftpobj,'MATLAB')
dir(ftpobj) % Veure documents del subdirectori

%% Pujar document/s
mput(ftpobj,'prova.txt');

%% Pujar documents que tinguin part del text en comú
mput(ftpobj,'prova*txt');
dir(ftpobj)% Veure documents del directori

%% Llegir contingut del document (si és text)
readme = fileread('prova.txt')

%% Descarregar fitxers del FTP
% mget(ftpobj,'prova.txt');

%% Tancar la connexió amb el FTP
close(ftpobj)
```

## 8.2. Accés remot control robot

### 8.2.1. Comandes bàsiques: Dashboard Server

Servidor de comandes d'escriptori que permet controlar de forma remota el robot enviant comandes simples a la interfície gràfica d'usuari (GUI) a través de la comunicació TCP/IP basada en comunicació Socket. El conjunt de comandes permeten modificar configuracions del robot, el seu estat, saber informació... Es poden trobar en la següent font [37]. L'inconvenient de la connexió a través d'aquests ports és que no es pot utilitzar el robot si està en mode remot (PC com a client, robot com a servidor).

```
%% Inici del codi
%% Connexió amb el host
host = '130.130.130.51';
port = 29999; % Port amb comandes bàsiques.
s = tcpip(host, port, 'NetworkRole', 'client');
t = tcpclient(host, port)
s.InputBufferSize = 221600;
fopen(s);
disp('Connectat!');

%Llegim la resposta del servidor.
Resposta=fscanf (s);

% Eliminem els espais entre les comes i el text rebut.
Resposta=strip(Resposta)

%% Enviament comandes al robot
flushinput(s)
Comanda ='Power off'
fprintf (s, strcat(Comanda, '\n')),

%%Llegim la resposta del servidor.
Resposta=fscanf (s);

% Eliminem els espais entre les comes i el text rebut.
Resposta=strip(Resposta)

%%Tanquem la connexió.
fclose(s);
delete(s);
clear s;
disp('Desconnectat');
```

### 8.2.2. Comunicació avançada: Sockets

Amb aquest tipus de comunicació canvia el rol entre client – servidor respecte les comunicacions anteriors i permet intercanviar informació entre el programa del robot i el programa del PC. En aquest cas el robot actua com a client i el programa del Matlab com a servidor. La complexitat de la

comunicació pot anar des d'enviar simples comandes (per exemple: canvi el valor d'una variable) fins a enviar les posicions completes, lectura de l'estat de les entrades i sortides, lectura sensors força...

L'exemple que es mostra a continuació permet escollir el tipus de tasca que realitzarà el robot, prèviament programada, a partir de la selecció per l'usuari en el Matlab. Per aquesta aplicació és necessari tenir dos programes, un en el robot on hi ha definides les accions per a cada tipus de tasca i la lectura del Socket, i un programa en el Matlab que permet a l'usuari escollir la tasca a realitzar.

A continuació es mostren els codis dels programes per ordres d'execució, primer el del Matlab i després el del robot. Una vegada executat el programa del Matlab, ens apareixeran instruccions en pantalla.

```
%% ----- Comunicació per Sockets amb robot UR3e -----

% Configurar IP robot i ordinador.
% Posar la IP de l'ordinador en el programa del robot.
% Posar la IP del robot en el programa del Matlab.
% Primer executar el programa del Matlab
% -----

%% Inici del codi
clear all % borrar variables

% Connexió amb el host
IP_robot = '192.168.199.128';
Port = 50; % Port de connexió.
s = tcpip(IP_robot,Port,'NetworkRole','server'); % Definim com a servidor i
iniciem la connexió.
disp('Arrencar programa robot');
fopen(s); % Obrir la comunicació.
disp('Connexió establerta!');

%% Exemple format comanda a enviar tipus float.
% fprintf(s,'(1.0)') % Tasca 1 Seleccionada
% fprintf(s,'(2.0)') % Tasca 2 Seleccionada
% fprintf(s,'(3.0)') % Tasca 3 Seleccionada

%% Aturar la connexió
% fclose(s);
```

```
Program
BeforeStart
'Establiment de la connexió entre client i servidor.'
open:=socket_open("192.168.199.131",50)
Loop open= False
Wait: 0.01
open:=socket_open("192.168.199.131",50)
PosicioInicial:=p[0,0,0,0,0,0]
tasca:=0
Robot Program
'Programa d'execució de tasques del robot.'
Wait: 0.1
If tasca=1
Wait: 0.01
MoveJ
Waypoint 3
```

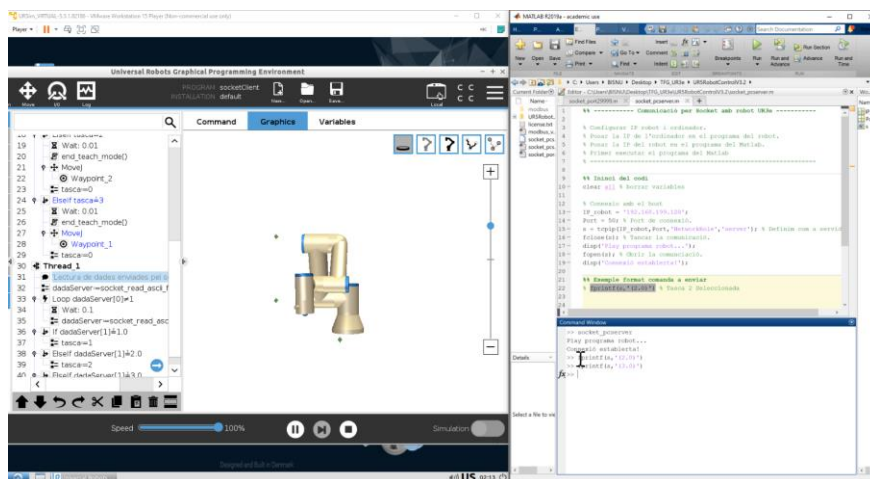
```

tasca:=0
ElseIf tasca=2
    Wait: 0.01
    MoveJ
        Waypoint_2
    tasca:=0
ElseIf tasca=3
    Wait: 0.01
    MoveJ
        Waypoint_1
    tasca:=0
Thread_1
    'Lectura de dades enviades pel servidor del tipus float.'
    dadaServer:=socket_read_ascii_float(1)
    Loop dadaServer[0]#1
        Wait: 0.1
        dadaServer:=socket_read_ascii_float(1)
    If dadaServer[1] = 1.0
        tasca:=1
    ElseIf dadaServer[1] = 2.0
        tasca:=2
    ElseIf dadaServer[1] = 3.0
        tasca:=3

```

Una vegada vist que és factible aquest tipus de comunicació i que funciona, en el següent capítol s'utilitza aquest mètode per posar en pràctica la comunicació en un cas particular on s'utilitza la visió per computador.

Per altra banda, s'ha elaborat un vídeo demostratiu del funcionament la comunicació a través d'una aplicació, el codi del qual és el que s'ha mostrat anteriorment.



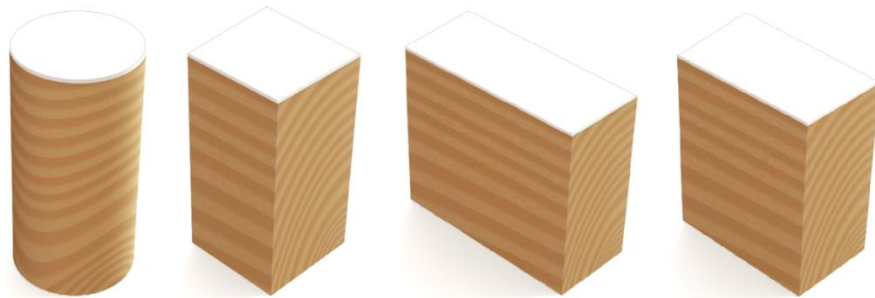
**Figura 8.1.** Captura vídeo demostratiu del funcionament de la comunicació (Font: Bisnu Masó)

Link vídeo: <https://youtu.be/s4PuFppHcrr>

## 9. Aplicació amb visió

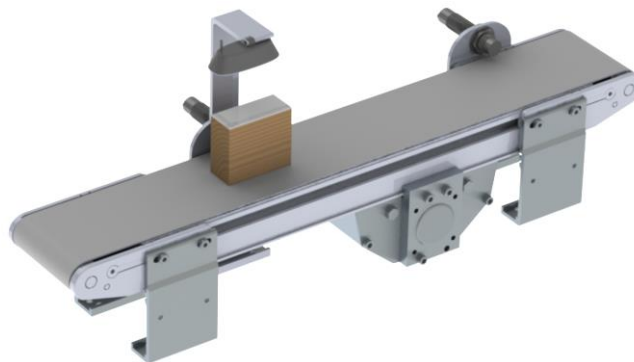
En aquesta secció es planteja una aplicació diferent que permet millorar la capacitat de l'estació i del robot. L'aplicació escollida és una aplicació amb visió per computador on s'utilitza el robot amb un sistema d'adquisició i processament d'imatges extern per analitzar unes figures i prendre unes decisions.

La tasca escollida podria ser una aplicació real de control de qualitat on s'analitza un tipus de peça en concret per detectar-hi defectes i excloure les peces que no passen els estàndards de qualitat. Per simplificar el concepte s'ha optat per escollir una aplicació que consisteix a classificar quatre tipus de peces diferents segons la seva geometria: circumferència, quadrat, rectangle petit i rectangle gran. En cas de no coincidir en cap d'aquestes quatre, la peça és descartada com a defectuosa (ex: peça amb superfície ovalada, peça sense geometria definida...).

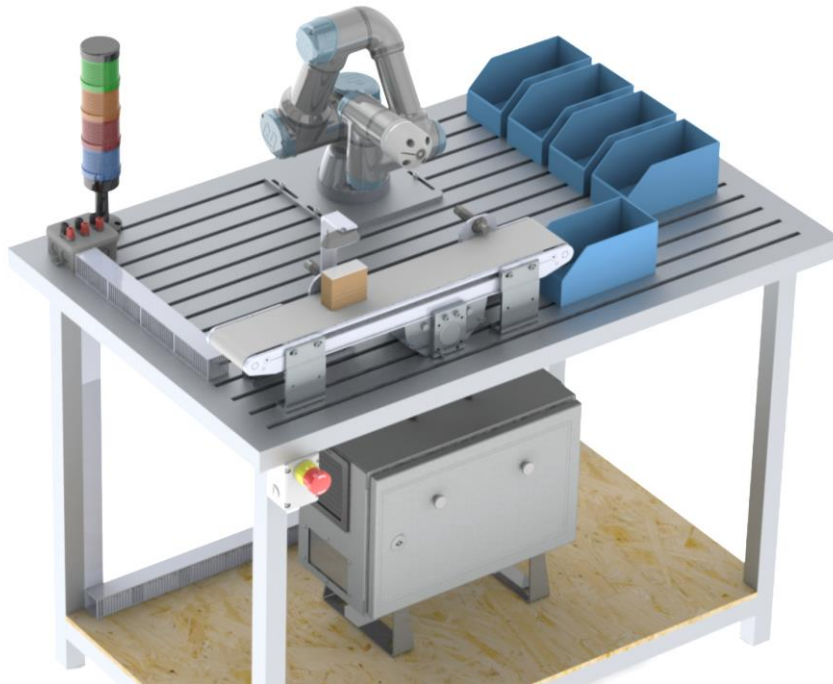


**Figura 9.1.** Disseny peces de diferents formes per la visió per computador (Font: Bisnu Masó)

Per realitzar aquesta part del treball s'ha adaptat l'estació amb els elements essencials: cinta de transport (en aquest cas s'ha escollit una de la marca Dorner Conveyors , model 2200), una càmera web per l'adquisició de les imatges i dos sensors inductius per posicionar les peces en unes posicions concretes. El muntatge del conjunt dels dispositius per dur a terme les tasques de visió per computador és la que es mostra a la següent figura:



**Figura 9.2.** Sistema de captura i transport de peces per l'aplicació de visió(Font: Bisnu Masó)



**Figura 9.3.** Estació completa per l'aplicació de visió per computador (Font: Bisnu Masó)

## 9.1. Funcionament

El funcionament és el que es mostra en el diagrama de blocs de la figura 9.4. Una vegada completats els passos de l'apartat 12.2 per configurar i establir la comunicació entre Matlab i URSim es procedeix en executar el programa del Matlab i, tot seguit, quan el programa ho indiqui, executar el programa del robot. Si el programa del Matlab notifica que s'ha establert la comunicació voldrà dir que el sistema està preparat per treballar i la cinta s'activarà a l'espera de posar-hi una peça.

Una vegada, a partir d'una acció manual de l'operari, s'ha introduït una peça, aquesta avança per la cinta fins que és detectada pel primer sensor inductiu que parará la cinta i enviarà un senyal al Matlab per iniciar l'adquisició de la imatge. Una vegada s'ha capturat la imatge s'envia un senyal al controlador per tornar a activar la cinta on la peça tornarà a avançar fins al següent sensor òptic on es parará esperant que el robot la reculli o sigui descartada. Paral·lelament amb el desplaçament de la peça, el programa del Matlab duu a terme el processament d'imatge. Una vegada acabat el anàlisi envia el senyal al robot per indicar el tipus de peça que ha analitzat i la tasca que s'ha de dur a terme. El robot, segons el valor rebut, recollirà la peça i la portarà en la caixa corresponent o activarà de nou la cinta per descartar la peça.

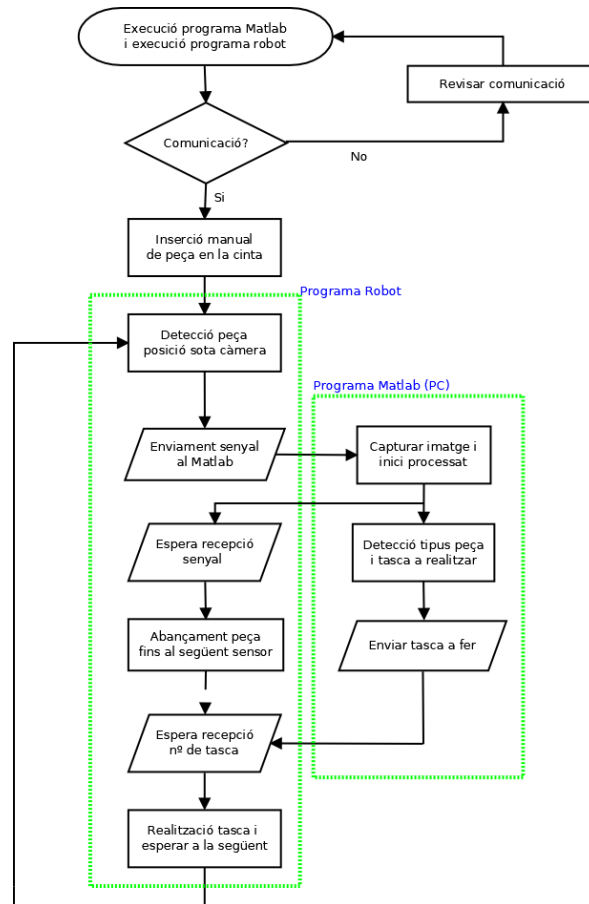


Figura 9.4. Diagrama funcionament aplicació de visió per computador (Font: Bisnu Masó)

## 9.2. Codi

Per elaborar els codis per aquesta aplicació s'han utilitzat les funcions detallades en l'apartat 8.2.2. *Comunicació avançada*: *sockets* donat que aquest tipus de comunicació és l'únic que permet enviar i rebre dades entre robot i Matlab.

A continuació es detallen els dos codis programats, el que s'executa en el Matlab i el que s'executa al robot.

### 9.2.1. Programa Matlab

El codi del Matlab és el que s'encarrega d'iniciar la connexió a través del *sockets* i el que dur a terme tota la part del processament de les imatges i permet diferenciar els tipus de peces que passen per sota la càmera a través d'una fotografia. El programa emprat per l'anàlisi d'imatges és el que es va programar en les pràctiques de l'assignatura de robòtica industrial i visió per computador (RIVC) i s'han dut a terme les modificacions pertinents per poder treballar amb el robot d'Universal Robots. Donat

que el codi de processament d'imatges no és rellevant ni la base del treball, aquesta part s'adjunta en els annexos. Per altra banda, per la part de demostracions s'han utilitzat imatges ja capturades i la lectura es fa a partir del fitxer i no d'una captura en temps real al no disposar de tots els equips per fer-ho directament sobre el robot.

```
%% Inicialització sistema de vídeo i port sèrie (OPCIÓ 2)
% Inici sistema de vídeo amb captura d'imatges en temps real. Els drivers poden
variar segons el model i resolució de la càmera.
% delete(imaqfind)
% vidobj = videoinput('winvideo', 1, 'YUY2_320x240');
% set(vidobj, 'ReturnedColorSpace', 'grayscale');
% triggerconfig(vidobj, 'manual');
% colormap(gray);
% start(vidobj);

%% ----- Comunicació per Socket amb robot UR3e -----

% Configurar IP robot i ordinador.
% Posar la IP de l'ordinador en el programa del robot.
% Posar la IP del robot en el programa del Matlab.
% Primer executar el programa del Matlab
% -----

%% Inici del codi
clear all % Borrar variables
clc

% Connexió amb el host
IP_robot = '192.168.199.128';
Port = 50; % Port de connexió.
s = tcpip(IP_robot, Port, 'NetworkRole', 'server'); % Definim com a servidor.
disp('Executa el programa del robot');
fopen(s); % Obrir la comunicació.
disp('Connexió establerta!');

%% Bucle principal del programa (s'executa de forma indefinida)
fin = 0;
while (fin==0)
    j=str2double(fscanf(s, '%c')); % S'espera la recepció del sensor inductiu 1
    per iniciar la seqüència de processat (s'espera rebre un 1.0 i es passa a 1)
    if j == 1

        %% (OPCIÓ 1)
        %% Lectura de la imatge des de fitxer extern.
        % fprintf('\n');
        str = input('Escriu el nom i extensió del fitxer: ', 's');
        % Es sol·licitarà posar el nom i extensió del document d'imatge.
        I = imread(str);
        fprintf(s, '(9.0) '); % Enviament senyal activació cinta
        j=0;

        %% (OPCIÓ 2)
        %% Captura imatge amb Webcam
        % Captura d'una imatge (foto)
        %I = getsnapshot(vidobj);
        % Visualització
        %figure, imshow(I);
        % Desconnexió entrada vídeo
        %stop(vidobj);
        %delete(vidobj);
    end
end
```



```

%% CODI DETERMINACIÓ DE LA PEÇA

%[aquí hi va el codi de processament d'imatges]

%% ENVIAR SENYAL AL ROBOT UNA VEGADA DETERMINADA LA PEÇA
switch (tipus_peca)
    case 1
        fprintf(s, '(1.0)'); % Tasca 1 Seleccionada
        disp('Peça detectada: quadrada');
    case 2
        fprintf(s, '(2.0)') % Tasca 2 Seleccionada
        disp('Peça detectada: rectangular petita');
    case 3
        fprintf(s, '(3.0)') % Tasca 3 Seleccionada
        disp('Peça detectada: rectangular gran');
    case 4
        fprintf(s, '(4.0)') % Tasca 4 Seleccionada
        disp('Peça detectada: ciruclar');
    otherwise
        fprintf(s, '(5.0)') % Tasca 5 Seleccionada
        disp('Peça detectada: defectuosa, descartar peça');
end

%[aquí hi va el codi per veure els resultats del processament
d'imatges]

end
end

%% Tancar el port sèrie (finalitzar comunicació PC - Controlador)
% fclose(s);
% delete(s);
% clear s;

```

### 9.2.2. Programa robot

El codi del robot és el que s'utilitza per definir les posicions i recorreguts del braç articulat i, alhora, permet enviar les lectures de les entrades i sortides dels ports del controlador a altres programaris que ho sol·licitin, en aquest cas, enviar la lectura dels sensors inductius al codi del Matlab.

El següent fragment de codi, corresponent a la primera part: **BeforeStart**, és el que permet poder establir una comunicació amb el Matlab a partir dels sockets. Aquest codi s'executa abans que el programa principal del robot i alhora permet establir els valors inicials de les variables.

```

Program
BeforeStart 'Part del programa que s'executa abans del programa del robot'
    'Establiment de la connexió entre client i servidor.'
    open:=socket_open("192.168.199.131",50)
    Loop open≠ False
        Wait: 0.01
        open:=socket_open("192.168.199.131",50)
    PosicioInicial:=p[0,0,0,0,0,0]

```

```

tasca:=0
Set DO[0]=Off
conveyorbit:=0 'Bit per a l'activació o apagada cinta.'

```

El segon fragment de codi corresponent al bloc **Robot Program**, és el bloc on incorpora totes aquelles accions que s'executen de forma indefinida, si no s'indica el contrari en el codi, una vegada s'ha iniciat el programa. En aquest cas, hi conté el programa i les seqüències de les diferents posicions des d'on ha d'anar a buscar la peça fins a on l'ha de deixar un cop ha rebut el senyal del tipus de peça `tasca=n`. Les tasques que comporten el moviment del robot és de la tasca 1 a la tasca 4 i l'última tasca només comporta l'activació de la cinta de transport.

#### Robot Program

```

'Programa d'execució de tasques del robot.'
Wait: 0.1
If tasca=0
    Wait: 0.01
    MoveJ
        PosEspera
    tasca:=0
ElseIf tasca=1
    tasca:=0
    Wait: 0.01
    Call Pick
    MoveJ
        PlacelUp
    MoveL
        PlacelDown
    Wait: 1.0
    Set DO[0]=Off
    MoveL
        PlacelUp
ElseIf tasca=2
    [codi]
ElseIf tasca=3
    [codi]
ElseIf tasca=4
    [codi]
ElseIf tasca=5
    tasca:=0
    conveyorbit:=0

```

El codi del bloc **Pick** correspon a un subprograma per anar a buscar la peça una vegada està parada en el segon detector inductiu. L'avantatge de generar un subprograma és que permet ser cridat tantes vegades com es vol i és de gran utilitat en seqüències on una part del codi és repetitiu, com en aquest cas on l'acció d'anar a buscar la peça a la cinta és la mateixa per a totes les 4 tasques definides en el bloc anterior.

```
Pick
  'Seqüència genèrica de recollida peça cinta'
  Wait DI[1]=HI
  MoveJ
    PickUp
  MoveL
    PickDown
  Set DO[0]=On
  Wait: 1.0
  MoveL
    PickUp
  conveyorbit:=0
```

Tant el bloc **Thread\_1** com el **Thread\_2** són blocs que, igual que el bloc del **Robot Program**, s'executen de forma indefinida però paral·lelament sense dependre d'altres parts del codi. Aquests blocs són útils per fer una sèrie d'accions que es volen que es vagin duent a terme de manera constant com per exemple lectura de dades dels ports de comunicació. En el cas del primer correspon a l'exemple esmentat, es va llegint de manera constant les dades que es reben pel socket i d'aquesta manera permet determinar la tasca rebuda encara que el robot estigui a mitja execució d'una altra part del programa. De la mateixa forma, en el **Thread\_2** permet tenir un control separat de la cinta de transport de manera independent.

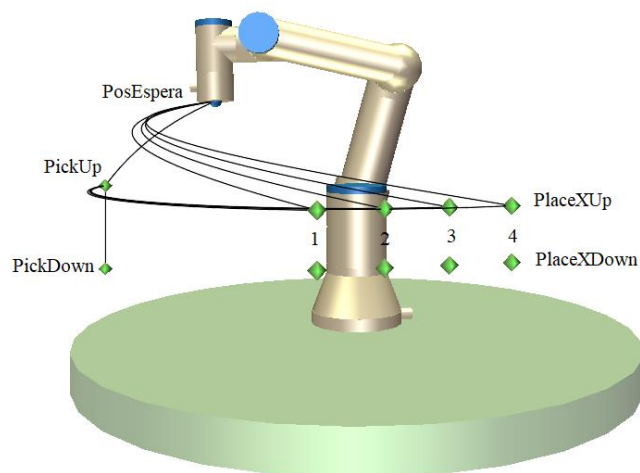
```
Thread_1
  'Lectura de dades enviades pel servidor.'
  dadaServer:=socket_read_ascii_float(1)
  Loop dadaServer[0]≠1
    Wait: 0.1
    dadaServer:=socket_read_ascii_float(1)
  If dadaServer[1]≧1.0
    tasca:=1
  ElseIf dadaServer[1]≧2.0
    tasca:=2
  ElseIf dadaServer[1]≧3.0
    tasca:=3
  ElseIf dadaServer[1]≧4.0
    tasca:=4
```

```

ElseIf dadaServer[1]≠5.0
    tasca:=5
ElseIf dadaServer[1]≠9.0
    conveyorbit:=1
Thread_2
    'Lectura sensor entrada, control cinta i enviament comanda al PC'
If digital_in[0]≠ True and conveyorbit≠0
        socket_send_line("1")
        socket_send_line("0")
        Set DO[0]=Off
ElseIf digital_in[1]≠ False and conveyorbit≠1
        Set DO[0]=On
ElseIf digital_in[0]≠ False and conveyorbit≠0
        Set DO[0]=On
ElseIf digital_in[1]≠ True and conveyorbit≠1
        Set DO[0]=Off

```

En la següent figura permet veure les posicions dels diferents punts de pas definits per a cada tasca.



**Figura 9.5.** Representació gràfica dels punts de pas i punts d'actuació aplicació de visió (Font: Bisnu Masó)

En el següent enllaç es pot veure la demostració del funcionament d'aquesta aplicació:

<https://youtu.be/ThPpd1TH2Wk>

## 10. Col·laboració humà-robot

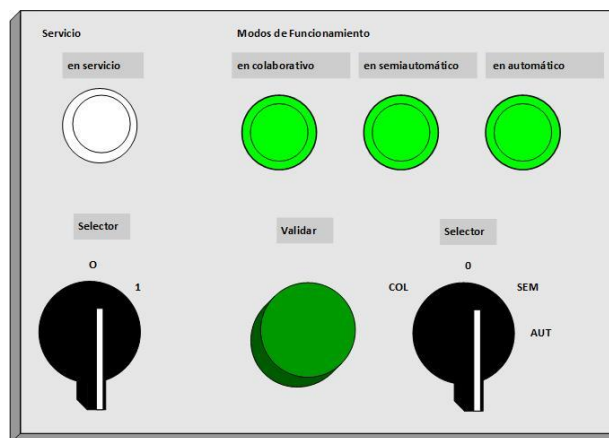
Per elaborar aquesta secció, amb col·laboració amb el professorat, es va escriure un article amb pautes pel disseny de tasques cooperatives humà-robot en àmbits acadèmics que aborden diferents aspectes a l'hora d'elaborar tasques humà-robot. Part del contingut, que tracta sobre un robot en general, sense definir cap model en concret, s'exposa a continuació.

La interacció entre humà i robot és una àrea emergent que s'està estenent amb l'aparició dels robots col·laboratius i la transformació del model de seguretat que s'aplicava fins a l'actualitat on no es contemplaven aquests tipus de robots.

La col·laboració humà-robot (HRC) comporta que ambdós individus comparteixen l'àrea de treball. Si les tasques es duen a terme de forma paral·lela, es considerarà que hi ha una cooperació paral·lela, en cas d'haver-hi una cooperació més activa entre humà-robot els quals treballen colze a colze, es considerarà una cooperació col·laborativa en la qual el risc de col·lisions és el més elevat. La normativa actual UNE-EN ISO 1028-1 (UNE 2016) defineix el concepte d'espai de treball, associat amb aquesta idea d'espai màxim i restringit, al costat de la possibilitat que robot i humà puguin realitzar tasques de forma simultània en l'anomenat espai de treball cooperatiu [38].

Per augmentar les funcionalitats de l'estació i potenciar la part col·laborativa, s'han dissenyat diferents elements que permeten la interacció entre màquina-humà.

En primer lloc s'ha dissenyat un panell de comandament on es té en compte la següent funcionalitat vertical: posada en servei (alimentació elèctrica de l'estació) i modes de funcionament. A nivell de distribució horitzontal, en la part superior es poden observar indicadors LEDs i en part inferior els controls amb selectors i pulsadors que permeten determinar quin mode de funcionament es vol dur a terme.



**Figura 10.1.** Disseny i distribució elements panell de comandament (Font: Bisnu Masó)

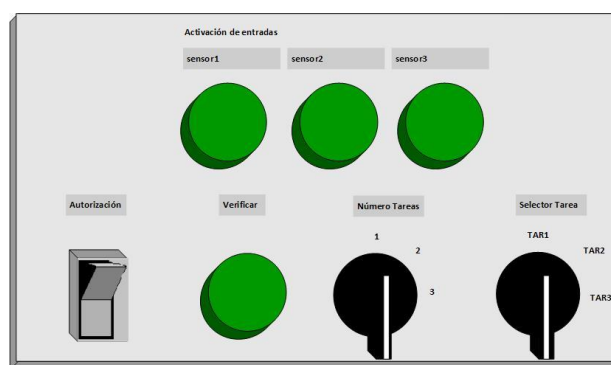
En segon lloc, s'ha dissenyat uns pilots lluminosos que permeten informar a l'operari de l'estat de l'estació en tot moment.

**Taula 10.1.** Descripció color indicador lluminós amb relació a l'estat de l'estació.

Color pilot	Estat estació
Verd	Funcionament normal.
Taronja	Alarma (avís robot finalitzant tasques automàtiques)
Vermell	Error de dispositiu, requereix parar l'estació.
Blau	Accés i/o control remot activat*

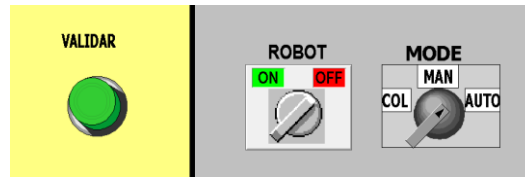
\*La normativa ISO 10218-2-2016 (ISO 20016) defineix l'accés remot amb intervencions manuals pel diagnòstic de robot.

En tercer lloc s'ha dissenyat un panell auxiliar d'entrades i sortides digitals que afegeix flexibilitat a l'hora de programar el robot si la perifèria del robot no està completada. En la part superior s'hi poden trobar pulsadors per activar entrades digitals, en la part inferior, d'esquerra a dreta, hi ha un interruptor d'autorització, un pulsador de verificació i dos selectors que permeten seleccionar la tasca i el nombre de tasques que es volen dur a terme.



**Figura 10.2.** Disseny i distribució panell auxiliar (Font: Bisnu Masó)

Una vegada elaborats els dissenys, si es vol aplicar aquests conceptes pel robot del laboratori i es fa el recompte del nombre total d'entrades necessàries, es tenen que pel primer panell són necessaris 4 entrades digitals i pel segon panell 5 entrades digitals. Considerant que la caixa de control del robot UR3e disposa de 8 entrades digitals s'observa que no es disposen de suficients ports d'entrades. Tot i l'existència de mòduls d'expansió d'entrades i sortides com el IOFIREBUG [39], al no disposar d'aquest element, es proposa una alternativa simplificada de les idees anteriors on només són necessaris quatre ports d'entrades:



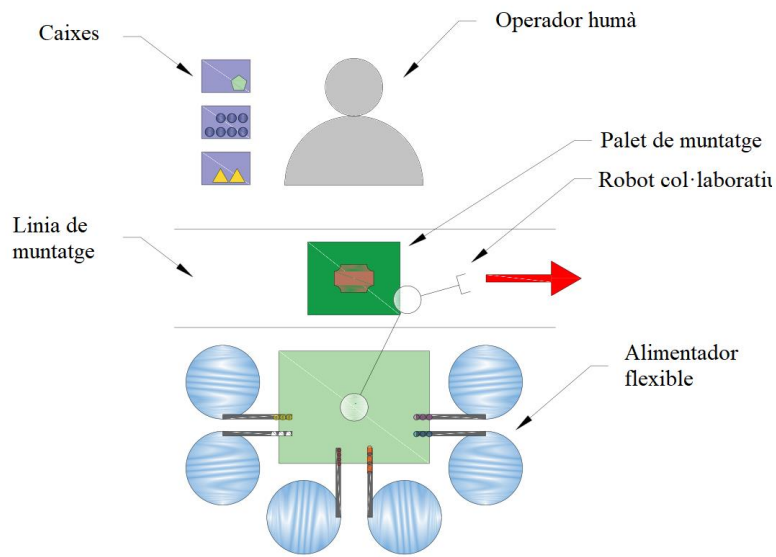
**Figura 10.3.** Disseny i distribució elements panell alternatiu (Font: Bisnu Masó)

El botó de l'esquerra permet executar confirmacions de selecció o notificació de la tasca finalitzada per l'operari donant accés al torn del robot per dur a terme les seves tasques. Els dos selectors que es troben en la part dreta del panell, el de l'esquerra permet habilitar o deshabilitar el robot (en cap moment es desconnecta la tensió d'alimentació del controlador del robot) i el de la dreta permet seleccionar el mode de funcionament. El que es pretén amb aquest selector és canviar paràmetres del robot en termes de velocitat, força i parell per adequar la seguretat segons el tipus de tasca que s'executa a l'àrea de treball.

## 10.1. Aplicacions col·laboratives

Si s'analitzen els usos dels robots col·laboratius en la indústria, es troben pocs casos on s'utilitzi per desenvolupar tasques colze a colze amb l'operari. La majoria de vegades s'utilitzen per als avantatges que ofereixen en terme de seguretat i la facilitat de programació. Al no haver de disposar de gàbies de seguretat, permet optimitzar millor els espais i automatitzar processos on les tasques es duen a terme per un operari donat que no hi havia la possibilitat d'incorporar un robot convencional. Les tasques que es solen dur a terme, el mode de cooperació solen ser la coexistència, la cooperació seqüencial i la cooperació paral·lela.

Una aplicació comuna on sí que existeix una cooperació directa, és els processos de muntatge que es duen a terme en els sistemes flexibles d'assemblatge (FAS, *Flexible Assembly Systems*), sistemes capaços d'adaptar-se al producte a elaborar de manera ràpida i flexible en comparació amb els tradicionals sistemes d'assemblatge manuals (MAS, *Manual Assembly Systems*). A l'incorporar un robot col·laboratiu en aquests sistemes reben el nom de CAS (*Collaborative human-robotAssembly*). En aquests casos el robot realitza les tasques més repetitives: col·locació de peces pesants, inserció de coles o greix en les juntes, cargolar els cargols, facilitar material a l'operari, etc. En canvi l'humà fa les tasques de verificació, col·locació de peces crítiques, supervisió de processos i totes aquelles activitats que no pot dur a terme un robot. Més enllà de tasques d'aquest tipus és difícil trobar una col·laboració directa.



**Figura 10.4.** Distribució sistema flexible d'assemblatge(FAS) (Font:[40])

En l'àmbit acadèmic, una aplicació de col·laboració humà-robot podria ser el muntatge de les peces de l'estació FAS200 on l'operari posa unes peces i el robot unes altres. Una aplicació més complexa i elaborada seria el joc del tres en ratlles amb la màquina. A partir de la visió artificial es podria crear un algorisme amb Matlab i jugar contra la màquina.

En el següent apartat del cas pràctic de les jornades es posar un exemple de col·laboració humà-robot on es dur a terme una tasca cooperativa paral·lela en la zona 3.



## 11. Cas pràctic: jornades portes obertes

Per les jornades de portes obertes del campus Diagonal-Besòs, Escola d'Enginyeria de Barcelona Est, que es va celebrar el 12 de febrer d'aquest any, es va exposar el robot en el vestíbul de la universitat en el departament d'electrònica industrial i automàtica per tal de donar visibilitat dels equipaments que es disposen en el centre.

### 11.1. Disseny i distribució

Per l'exposició es va dissenyar un conjunt de tasques per tal de mostrar les funcionalitats del robot col·laboratiu. En primer lloc, es va plantejar una distribució de la taula per distribuir-hi les accions que s'han de dur a terme. Tal com es mostra en la següent figura es van delimitar tres zones bastant diferenciades segons el tipus de seguretat i interacció amb el robot:

- **Zona blava (1):** espai on es du a terme una tasca repetitiva sense supervisió de cap persona o operari de forma automàtica pel robot. L'operari no hi ha de dur cap classe de tasca en aquesta zona.
- **Zona verda (2):** espai on es desenvolupa una tasca de forma automàtica amb interacció amb l'operari però mai al mateix moment. Una vegada el robot acaba la tasca, espera el senyal de l'operari per tornar a iniciar el programa. Útil quan l'operari és l'encarregat de proveir o recollir les peces al final del procés.
- **Zona vermella (3):** espai on se situen les tasques on intervé l'operari, la caixa de botons i pantalla de control (*teach pendant*). Aquesta és la zona més crítica a l'haver-hi una interacció humana directament amb el robot i existir la possibilitat de col·lisió entre humà-robot en ser una zona de treball compartida.

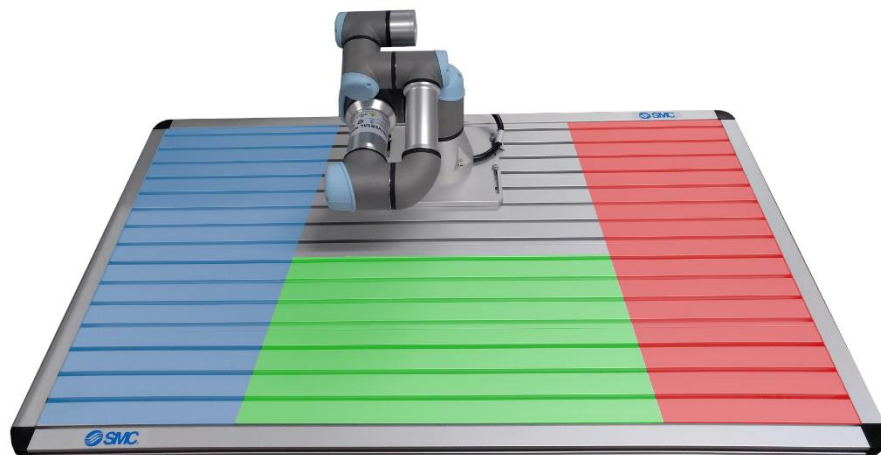
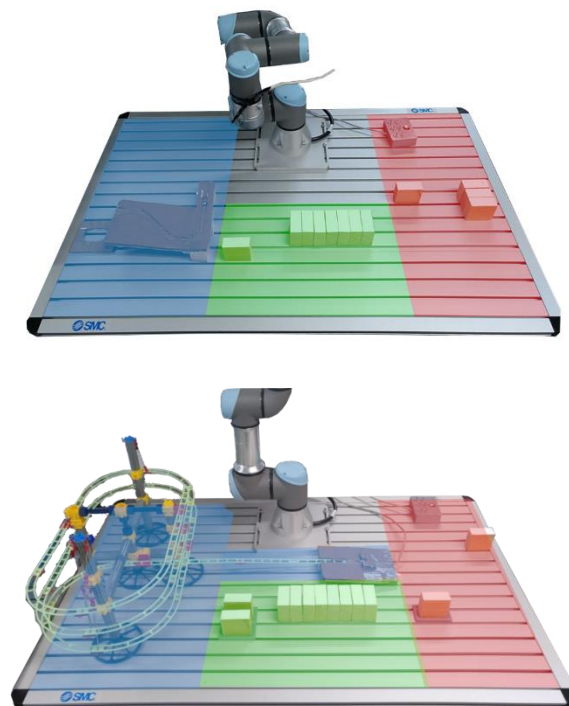


Figura 11.1. Distribució àrea de treball amb les tres zones (Font: Bisnu Masó)

Una vegada definides les zones es varen plantejar tres tasques diferents per dur a terme en cada una de les zones i es va elaborar un prototip de peces i estacions per fer les diferents proves:

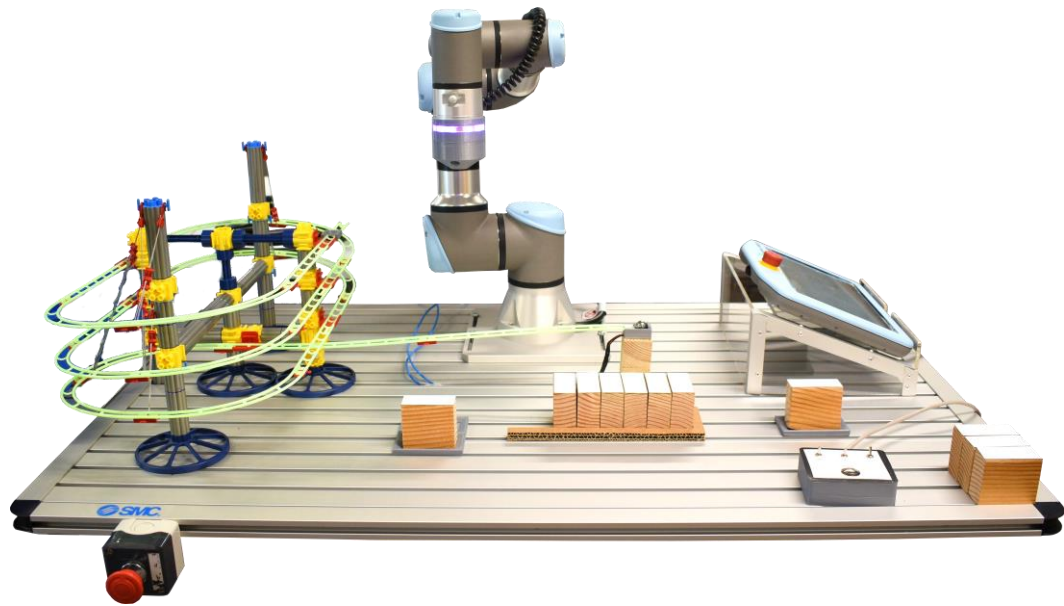
- **Zona 1:** format per una tasca automàtica on hi ha una estació que consisteix en un joc per a bales, en aquest cas de ferro, que recórrer un circuit muntat quan el robot diposita la bala en la part superior on hi ha la via de baixada.
- **Zona 2:** format per una tasca automàtica amb interacció humana basada en un programa de “despaletització” i paletització on el robot agafa peces de fusta, amb una superfície magnètica d’una posició i les col·loca en un altre lloc i estructura diferent, concretament en un palet de 1x2x3 (tres pisos de 2 peces en cada un).
- **Zona 3:** format per una tasca manual-automàtica on l’operari deixa una peça en una base i el robot la va a buscar i la col·loca en el nou palet.

Els primers prototips d’estació són els que es mostren en les següents figures. En la figura superior, el primer prototip, es va dissenyar una rampa per la bala amb cartró i més endavant, en la figura inferior, ja s’hi van incorporar les guies definitives amb un prototip de sensor per la bala.



**Figura 11.2.** Primeres proves de disseny d’estació i prototips (Font: Bisnu Masó)

El disseny final de l’estació és el que es mostra en la següent figura on s’hi poden observa les 3 tasques diferents, el panell *teach pendant* situat en la part superior dreta de la taula, una botonera en la part inferior dreta per seleccionar la tasca i un botó de parada d’emergència situat en la part inferior esquerra.



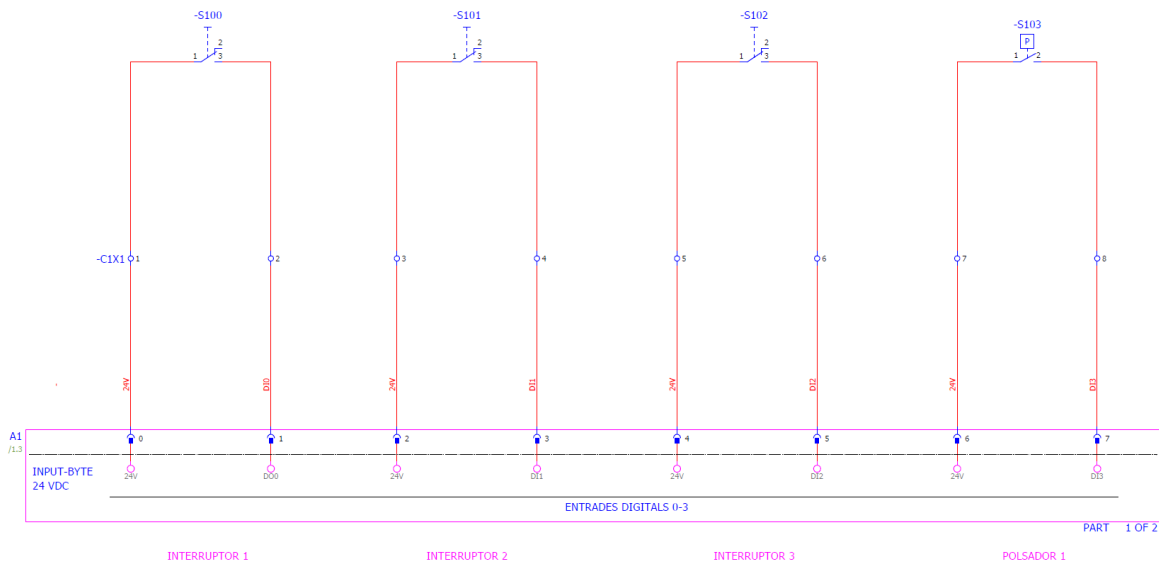
**Figura 11.3.** Disseny final d'estació amb les tres tasques diferents (Font: Bisnu Masó)

## 11.2. Funcionament

Una vegada el programa s'executa des de la consola de programació, a partir de la següent botonera permet iniciar o parar les tasques i triar quina tasca es vol que es dugui a terme. La botonera està formada per tres interruptors i un pulsador normalment obert connectats als ports d'entrada digitals del controlador del robot (figura 11.5).



**Figura 11.4.** Botonera selecció de tasca i zona de treball (Font: Bisnu Masó)



**Figura 11.5.** Esquema elèctric i ports de connexió botonera (Font: Bisnu Masó)

La seqüència del funcionament és el següent: una vegada posat en marxa el programa des de la consola, la resta d'accions es duen a terme a partir de la botonera. Amb el primer interruptor permet iniciar o posar en pausa les tasques. Una vegada el botó està en mode ON, a través de la resta d'interruptors i el botó, permeten seleccionar la tasca a executar:

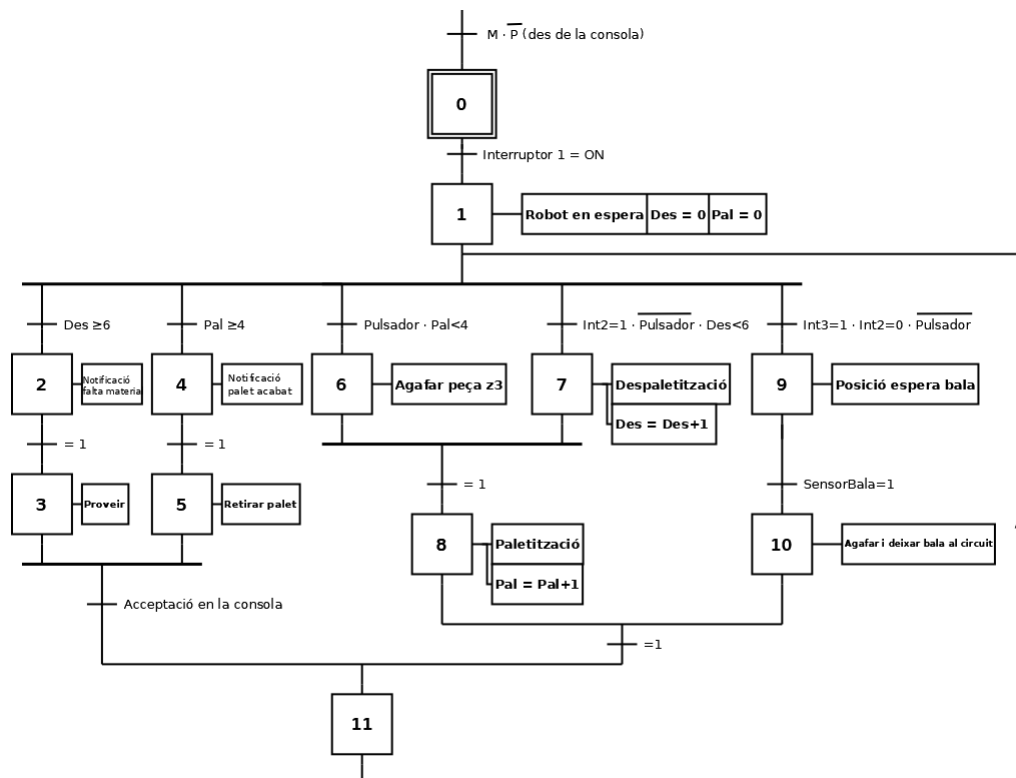
- **Polsador:** quan l'interruptor 1 està en ON i es prem el polsador, indiferentment de l'estat de la resta d'interruptors, el robot passa a dur a terme la tasca col·laborativa on agafa la peça que l'hi ha deixat l'operari i la porta en l'estació de paletització. Una vegada finalitzada la seqüència, si no torna a ser premut, continuarà en la tasca que estava fent abans de la interrupció.
- **Interruptor 2 (Int2):** quan està en la posició ON i el polsador no està premut s'executa la seqüència de despalatització on el robot agafa les peces de l'estació central i les porta en l'estació de paletització.
- **Interruptor 3 (Int3):** quan està en la posició ON i la resta en OFF, excepte el primer, el robot passa a la tasca automàtica on agafa una bala de ferro situat al final d'estació i la situa al cim de les vies per on baixarà per força de la gravetat fins a arribar una altra vegada en el sensor i tornar a iniciar el procés.

De forma paral·lela, el propi programa incorpora dos comptadors: un comptador que conta el nombre de peces que es col·loquen en el palet "*pal\_counter*" i un comptador que conta el nombre de peces que es treuen de la zona de despalatització "*des\_counter*". D'aquesta forma, una vegada el paleta està completat o no hi ha peces per paletitzar, el robot es mou en una posició d'espera i envia una

notificació en forma de “Popup” indicant que s’ha completat el palet o que falta material en la zona de despaletització.

### 11.3. Programa

Per la millor comprensió d’aquest programa s’ha elaborat un grafcet de nivell 1 que és mostra en la següent figura 11.6 i alhora és una ajuda a l’hora d’elaborar el programa del robot.



**Figura 11.6.** Grafcet de nivell 1 del funcionament del programa (Font: Bisnu Masó)

El programa s’ha elaborat a partir de la interfície de programació de la consola o PolyScope donat que són un conjunt de tasques senzilles sense complexitat. El programa està format per un programa principal i tres subprogrames que s’executen quan són cridats en el programa principal. El codi obtingut és el que es troba a continuació. (NOTA: s’han eliminat algunes línies de codi que no són rellevants per la seva comprensió).

```
// Fitxer .txt generat vinculat amb el fitxer del programa .urp
Program // Engloba tot el codi del programa.
  Init Variables // Es declaren els valors inicials de les variables.
  BeforeStart // Declaració valors de les variables quan s’executa el
programa
```

```

pal_counter=0
des_counter=0
Set DO[0]=Off
MoveJ
  Waypoint_2
Robot Program // Codi que s'executa de forma contínua.
If digital_in[3]≠ True and digital_in[0]≠ True //Tasca zona 3
  MoveJ
    Waypoint_1
  MoveL
    If force()≤3 // Si la força es major de 3N, atura el moviment.
      Waypoint_3
      Wait: 0.75
      Set DO[0]=On
      Wait: 0.75
    MoveL
      Waypoint_4
      Call paletitzacio_jor
  ElseIf digital_in[0]≠ True and digital_in[1]≠ False and
digital_in[3]≠ False and digital_in[2]≠ True //Tasca zona 1
    Call automatic_jor
  ElseIf digital_in[0]≠ True and digital_in[1]≠ True and
pal_counter≤3 and des_counter≤5 //Tasca zona 2
    Call despaletitzacio_jor
    Call paletitzacio_jor
  ElseIf digital_in[0]≠ False //Mostra notificacions "Popup"
    MoveJ
      Waypoint_2
      Popup: Prem Boto per iniciar
  ElseIf des_counter≥6
    MoveJ
      Waypoint_2
      Popup: Falta Material Alimentador. Repostar
      des_counter=0
  ElseIf pal_counter≥4
    pal_counter=0
    MoveJ
      Waypoint_2
      Popup: Palet Completat. Fer un altre palet?

// Subprogrames declarats
automatic_jor
despaletitzacio_jor
paletitzacio_jor

```

Per poder demostrar el correcte funcionament, s'ha elaborat un vídeo on es mostren les diferents tasques que realitza el robot i el funcionament general de l'estació: <https://youtu.be/ThPpd1TH2Wk>

## 12. Material d'ajuda a la docència

En aquest capítol s'inclou el contingut elaborat explícitament en forma de manual d'ajuda o codis com a plantilles que poden ser útils per a la docència.

### 12.1. Instal·lació màquina virtual i URSim

El programa URSim és el software de programació offline i simulació dels robots d'Universal Robots. Aquest programa està basat amb el sistema operatiu Debian i s'ha d'executar en una màquina virtual en el cas de voler-ho executar en un ordinador amb sistema operatiu Windows o iOS. Els passos per instal·lar el simulador es descriuen a continuació:

**Pas 1:** Descarregar i instal·lar una màquina virtual (VMware o VirtualBox). Es recomana el VMware (VMware Workstation Player) en ser el mateix que està instal·lat en els ordinadors del laboratori.

Link: <https://www.vmware.com/products/workstation-player.html>



Figura 12.1. Icona VMware (Font: Bisnu Masó)

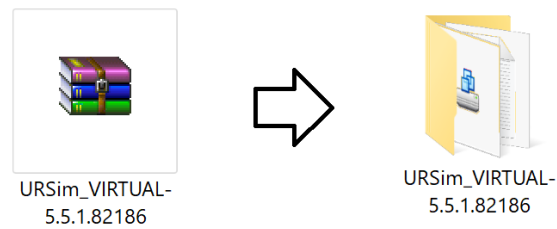
**Pas 2:** Descarregar el simulador des de la web de descàrregues oficial d'UR. Seleccionar la versió 5.5.1 que és la que conté el robot. D'aquesta manera no hi haurà discrepàncies de funcions.

Link: <https://www.universal-robots.com/download/>

<b>1. Select robot type</b> Label on the robot arm and control box will show what type it is.	e-Series
<b>2. Select type of download for e-Series</b> Search for article "How to use this Support site" for more information	Software
<b>3. Select type of software</b> Robot software should ONLY be installed by trained personnel. The software is not backwards compatible.	Offline Simulator
<b>4. Select operating system for your computer</b> URSim is a simulation software intended for offline programming and simulation of both robot programs and manual movement of robot.	Non Linux
<b>5. Select Software version</b>	UR Sim for non Linux 5.5.1

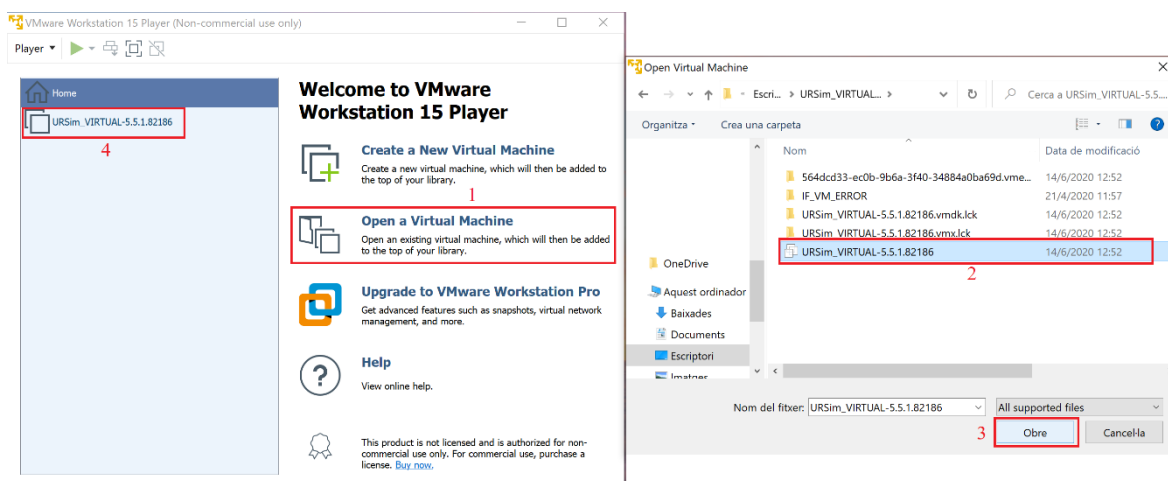
Figura 12.2. Menú selecció per descarregar URSim (Font: Bisnu Masó)

**Pas 3:** Una vegada descarregat l'arxiu que es troba al final de la pàgina, es descomprimeix la carpeta per tal que la màquina virtual pugui llegir els fitxers:



**Figura 12.3.** Descompressió carpeta descarregada (Font: Bisnu Masó)

**Pas 4:** Executar la màquina virtual instal·lada > fer clic a *Open a Virtual Machine* > Anar a la ubicació de la carpeta descomprimida anteriorment > Seleccionar el fitxer tal com es mostra en el requadre 2 > Clicar a obrir (3). Una vegada efectuat aquest pas, es podrà accedir a la màquina virtual amb la icona que s'ha creat (4).



**Figura 12.4.** Captura instal·lació URSim a VMware (Font: Bisnu Masó)

## 12.2. Creació de programes

Per crear un programa és elaborar directament en la consola de programació o en el simulador URSim. En els dos casos el procediment és el mateix. A continuació es detallen els passos per elaborar els programes i es posa un exemple d'un programa senzill de demostració.

Una vegada es té obert el programa del robot en el simulador o el Polyscope, per començar un nou programa s'ha d'excedir en la pestanya *Program*.



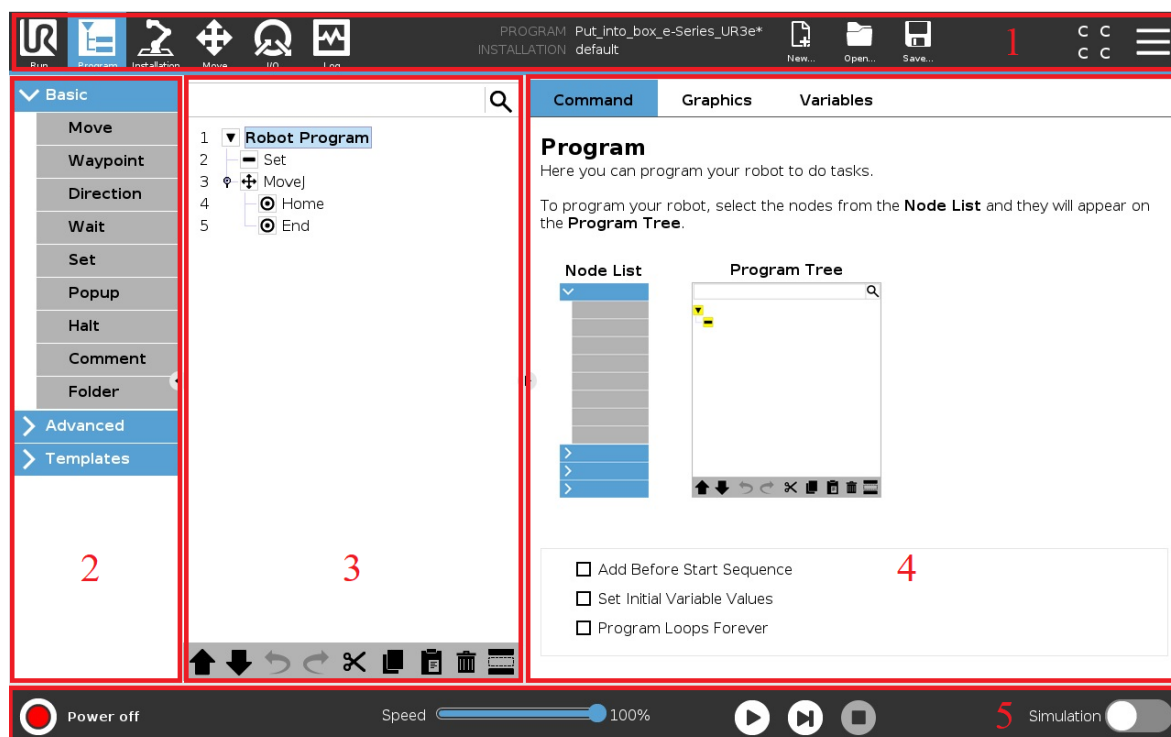


Figura 12.5. Captura pantalla de programació UR (Font: Bisnu Masó)

Tal com es pot veure en la pantalla de programació de la figura anterior, aquesta es pot dividir en cinc espais bastant diferenciats. En la barra superior, espai núm. 1, s'hi troben les opcions per navegar entre les diferents pantalles: pantalla inicial, pantalla de programació, pantalla de configuració del robot, pantalla per moure el robot, la pantalla d'entrades i sortides i la pantalla dels registres dels diferents successos. Per altra banda amb les icones situades a la dreta permeten navegar entre els diferents programes i gestió de l'emmagatzematge d'aquests mateixos.

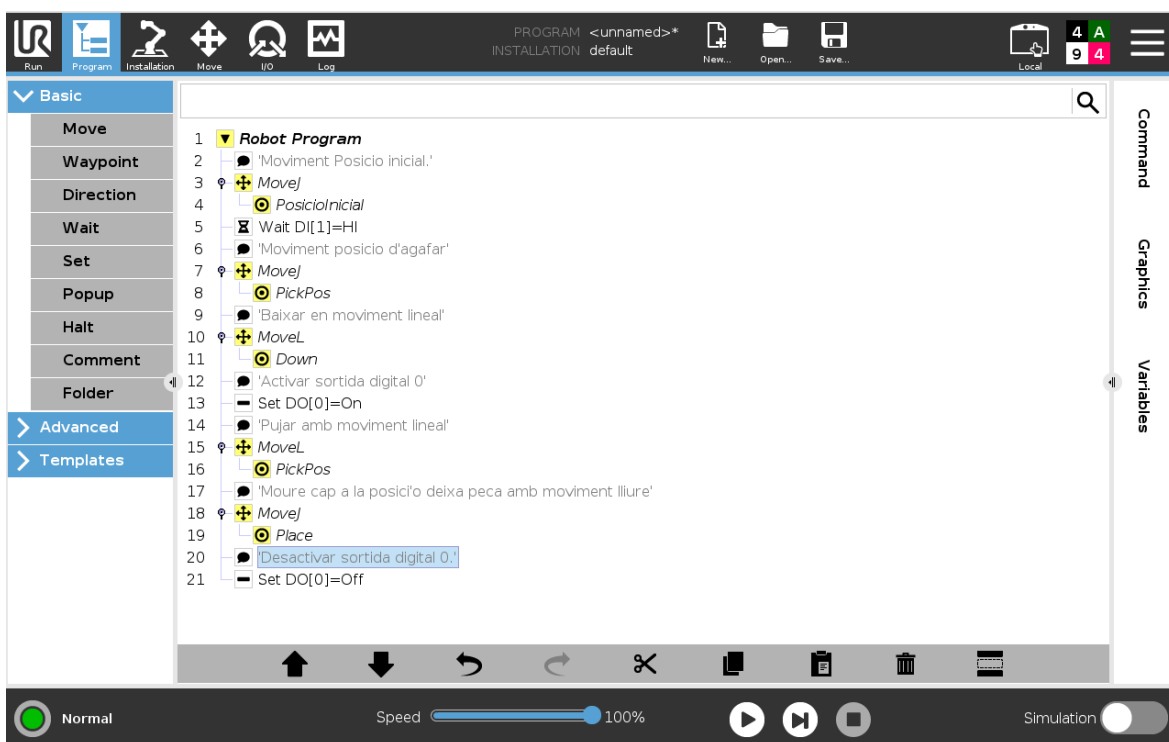
En el requadre de l'esquerra, espai núm. 2, s'hi troben les diferents comandes per programar el robot. Aquestes comandes estan dividides en bàsiques, avançades i plantilles, on s'hi troben diferents estructures i funcions ja definides per facilitar la programació. En l'espai núm. 3 és on es visualitza la seqüència de codi del programa i permet editar les diferents línies de programació. L'espai núm. 4 és la part de la pantalla que facilita ajuda durant la programació i indica com configurar cada una de les comandes que s'utilitzen i, alhora, dur a terme configuracions de velocitat, de força, veure els punts de referència (*waypoint*) i les diferents variables utilitzades pel programa. La barra de la part inferior, espai núm. 5, permet encendre o apagar el robot, que consisteix a alimentar o treure l'alimentació dels actuadors i activar o desactivar els frens de seguretat. També permet modificar la velocitat del robot i iniciar o aturar l'execució del codi programat en l'espai número 3.

**Taula 12.1.** Descripció de comandes per la programació intuïtiva.

Comanda	Descripció
<b>BASIC</b>	
Move	Comanda per definir el tipus de moviment entre els diferents punts de pas ( <i>Waypoints</i> ). <ul style="list-style-type: none"> <li>- MoveJ: Moviment lliure entre dos punts o més de pas.</li> <li>- MoveL: Moviment lineal entre dos punts o més de pas.</li> <li>- MoveP: Moviment circular entre dos punts o més de pas</li> </ul>
Waypoint	Definir els diferents punts de pas que conformen un o varis moviments.
Direction	Moviment lineal relatiu respecte la funció definida. (Només es pot utilitzar en els MoveL i MoveP.
Wait	Temps d'espera o lectura d'un senyal per passar a la següent comanda.
Set	Definir o forçar variable a un valor determinat.
Popup	Mostra un missatge en format <i>Popup</i> (davant de qualsevol pantalla oberta).
Halt	Para l'execució del programa en el punt definit.
Comment	Permet afegir comentaris entre les línies de codi de programació.
Folder	Conjunt de línies de codi agrupats en una carpeta.
<b>ADVANCED</b>	
Loop	Comanda per crear bucles i repeticions de part parcials del programa.
SubProg	Permet triar i cridar subrutines o programes externs a l'actual.
Assignment	Permet assignar una expressió matemàtica o comanda Script en una variable.
If	Condició que s'activa depenent de l'estat del sensor o variable de programa
Script	Entrada manual de codi de programació com a Script que executa el controlador de robot.
Event	Interrupció que s'activa segons l'estat d'un sensor o una variable i que no te efecte a la resta del programa mentre la seqüència s'està executant.
Thread	Seqüència de programa que s'executa de forma paral·lela amb el programa principal del robot.
Switch	Permet executar més complexes del tipus de condicions segons el valor d'una variable del tipus <i>case nomvar is</i> .
Timer	Temporitzador que conta el temps i apareix en una variable.
Screwdriving	Funció plantilla per dur a terme tasques de cargolar i descargolar cargols.
Home	Permet moure a la posició definida com a <i>Home Position</i> en la configuració del robot en la pestanya <i>Installation</i> .
<b>TEMPLATES</b>	

Seek	Plantilla que permet crear programes d'apilament d'objectes de diferents magnituds en alçada.
Force	Plantilla que permet crear programes on s'utilitza el sensor de força per detectar forces i prendre decisions.
Palletizing	Plantilla per crear seqüències de paletització i despaletització de forma efectiva i eficaç.
Conveyor Tracking	Plantilla que permet seguir el moviments de la cinta i dur a terme seqüències de programes de forma síncrona seguint el desplaçament de la peça.

Una vegada conegudes les línies bàsiques de programació, es crea un exemple senzill que consisteix a agafar una peça d'una posició, quan el sensor indica que hi ha peça, i el deixa en una altra posició que podria ser una caixa. El codi quedaria de la següent forma i els comentaris de cada pas es troben en el mateix codi:



**Figura 12.6.** Programa d'exemple d'un programa senzill. (Font: Bisnu Masó)

Per moure el robot a la posició que es vol es pot dur a terme de tres formes diferents. La primera forma és a partir dels controls situats en la part esquerra de la pantalla de *Move*. La segona manera (només habilitat en un robot físic) és amb l'opció de moviment lliure on l'humà és qui mou i situa el robot a la posició desitjada. Per habilitar aquest moviment s'ha de tenir premut el botó inferior de la pantalla que posa *Freedrive* o el botó que es troba en la part posterior de la consola de programació. La tercera

manera de posicionar el robot és a partir de la introducció manual de les posicions en graus de cada una de les articulacions. Igual que si es fa en un robot físic o en el simulador, els plans de seguretat continuaran actius i el robot no podrà sobrepassar aquests plans. Exemple en la figura següent.

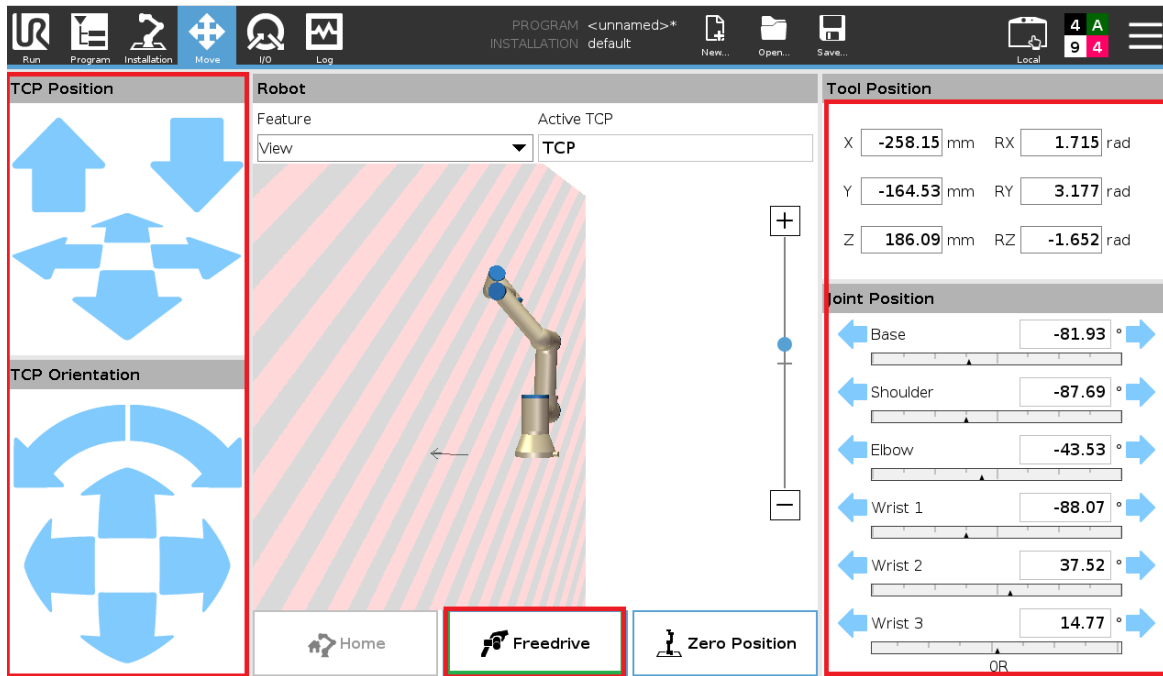


Figura 12.7. Mètodes per posicionar el robo (Font: Bisnu Masó)

### 12.3. Configuració comunicació URsim (VM)

A continuació s'explica com configurar la comunicació del simulador del robot, URsim, que alhora pot ser utilitzada per establir la connexió entre un software i el robot físic. En aquest cas es fa èmfasi en la part virtual.

Per establir una comunicació entre l'ordinador i el robot físic, l'usuari és qui escull l'adreça IP, la defineix en l'apartat de *Static Adress* (adreça estàtica, no varia) de configuració del robot i en l'adaptador de xarxa del mateix ordinador assignant una IP del mateix rang entre aquests dispositius. En el cas d'utilitzar el simulador del robot URsim, qui determina la IP a utilitzar és el mateix simulador que no permet establir una adreça estàtica sinó que ja ve donada a partir de protocol DHCP (*Dynamic Host Configuration Protocol*). Per saber quina és la IP definida en el robot s'accedeix a l'apartat de *Network* (*Settings > System > Network*) tal com es mostra en la següent figura.

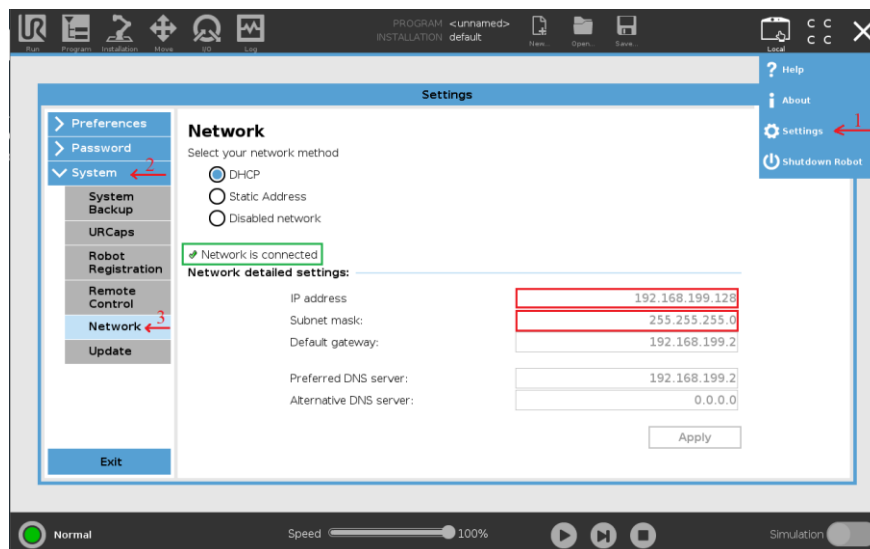
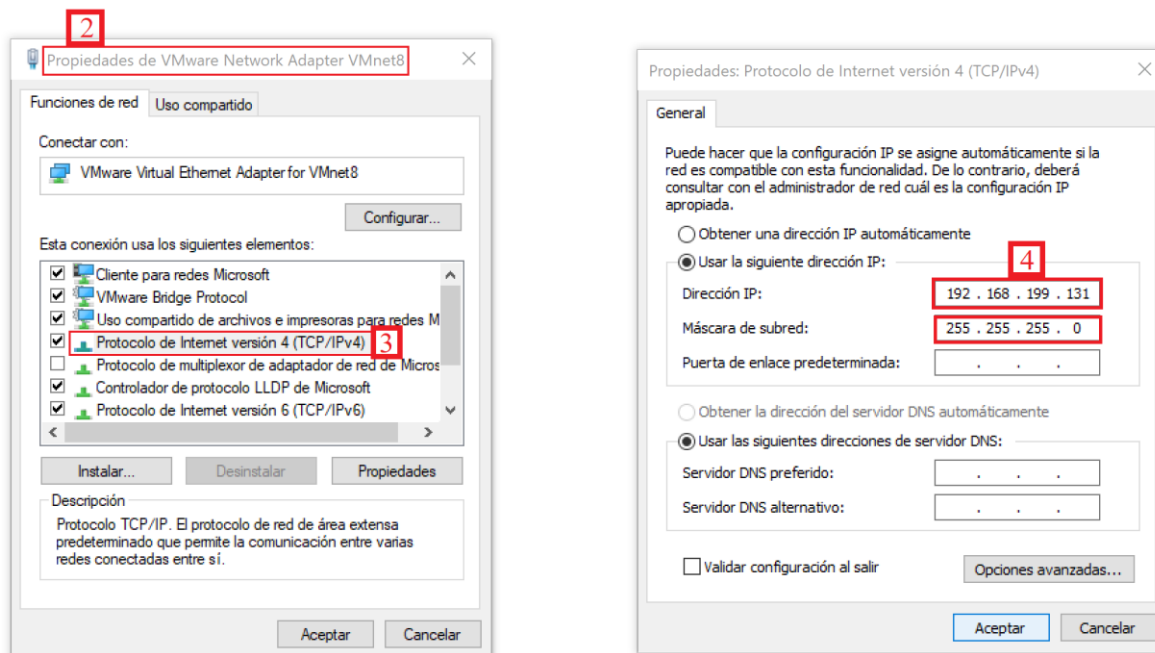


Figura 12.8. Captura configuració xarxa i IP per la comunicació (Font: Bisnu Masó)

Una vegada coneguda la IP de treball, es configura l'adaptador de xarxa. En tractar-se d'una connexió entre l'ordinador i una màquina virtual en el mateix dispositiu, el mateix programa de gestió d'aquestes màquines, en aquest cas VMware, crea uns adaptadors virtuals d'interconnexió entre aquests dos dispositius. Com que la màquina virtual està configurada per defecte l'adaptador VMnet8, es configura aquest mateix seguint les indicacions que es mostren en la següent figura. El protocol on s'ha de determinar una IP del mateix rang és el Protocol d'Internet versió 4 (TCP/IPv4). Alhora, també cal definir la mateixa adreça de la màscara de xarxa (*subnet mask*).

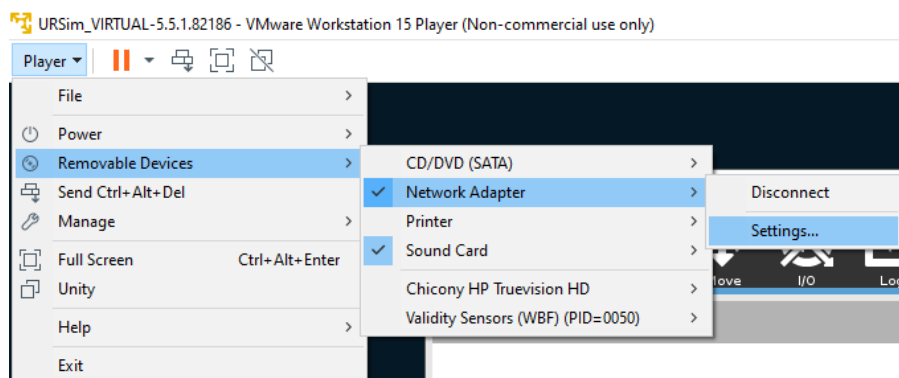


Figura 12.9. Captura adaptadors de xarxa ordinador (Font: Bisnu Masó)



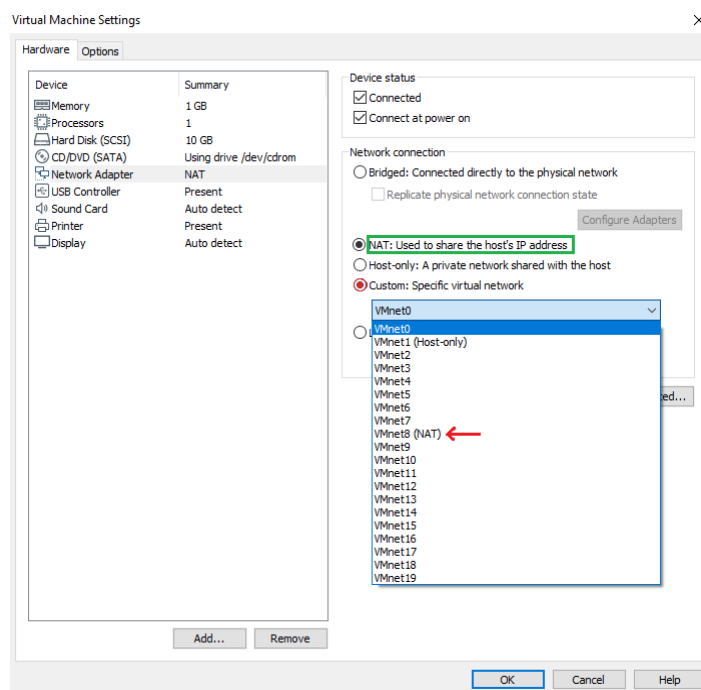
**Figura 12.10.** Captura paràmetres de configuració adaptador de xarxa ordinador (Font: Bisnu Masó)

En cas de desconèixer l'adaptador o no poder establir connexió, cal revisar quin és l'adaptador que està utilitzant la màquina virtual. Per saber quin és es pot accedir des de la configuració del VMware tal com es mostra en la següent figura.



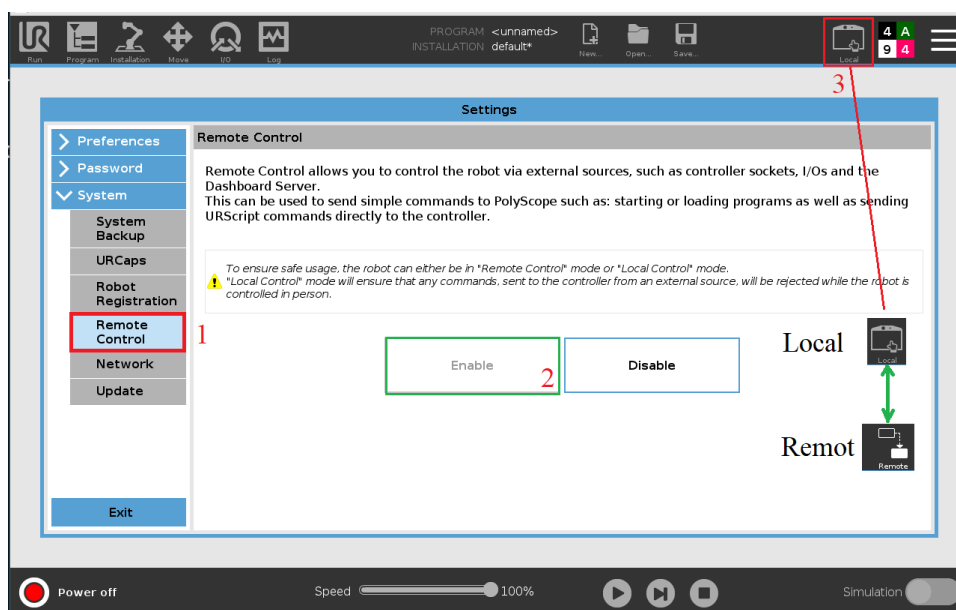
**Figura 12.11.** Captura accés configuració d'adaptador de xarxa màquina virtual (Font: Bisnu Masó)

En la configuració la connexió NAT vindrà seleccionada per defecte. Per veure a quin adaptador correspon es pot mirar en el desplegable de l'opció *Custom* (personalitzat) de la figura 12.9. En cas que es vulgui canviar, també es podrà fer però configurant correctament l'adreça IP tal com s'ha citat anteriorment.



**Figura 12.12.** Captura configuració adaptador de xarxa màquina virtual (Font: Bisnu Masó)

Una vegada configurada la IP ja es pot iniciar la comunicació entre dos dispositius on un d'aquest es defineix com a client i l'altre com a servidor. En el cas d'utilitzar el robot, caldrà seguir els mateixos passos i a més a més activar l'accés al mode remot: *Settings > System > Remote control > ENABLE*. Quan s'activa el mode remot, és possible canviar de mode sols fent clic al nou icona que apareix a la part superior dreta de la pantalla com indica el pas 3.



**Figura 12.13.** Captura habilitació mode remot robot físic (Font: Bisnu Masó)

## 12.4. Transferència programes entre USB-URsim i viceversa.

Una vegada instal·lat el programa URsim es te l'avantatge de poder utilitzar la programació offline i tot seguit, passar-ho al robot per veure el funcionament real del programa elaborat. El pas del programa entre el simulador i el robot es fa a través d'una memòria USB. Per tal de mostra el procediment a dur a terme, en aquest cas, s'han elaborat dos videotutorials. Un que es visualitza la transferència de fitxers des d'un *pendrive* al simulador i el segon que mostra el procés invers.

Link videotutorial 1: [https://youtu.be/2sTnGJv0\\_8g](https://youtu.be/2sTnGJv0_8g)

Link videotutorial 2: <https://youtu.be/tbSoXspQRR>



## Conclusions

Si s'analitzen els objectius proposats i els resultats obtinguts, una vegada realitzat el treball, es pot concloure que aquests s'han assolit de manera satisfactòria però no de forma completa. Tot i que actualment l'estació és operativa i funcional, no s'ha pogut incorporar la placa de terminals proposada en el capítol 5.3 perquè no s'ha pogut accedir al laboratori en aquests últims quatre mesos. Tota la part del muntatge i instal·lació que hi ha actualment es va dur a terme entre el gener i principis de febrer per tenir-ho preparat i poder exposar l'estació a les jornades de portes obertes del 12 de febrer.

Per altra banda, a mesura que s'anava realitzant el treball, s'han pogut detectar alguns inconvenients tant pel que fa a nivell software com de hardware. A nivell de software, en el mode de simulació, és necessari disposar d'una pestanya per a activar i desactivar els senyals d'entrades i sortides sense haver de canviar de pestanya de navegació entre *Program* i *I/O (Input / Output)*. El fet d'haver de canviar de pestanya, no és possible veure els moviments del robot donat que es perd la interfície gràfica del robot en la pestanya *I/O*. A nivell de hardware s'ha trobat a faltar la possibilitat de poder afegir-hi blocs d'expansió directament a la caixa de control donat que el nombre d'entrades i sortides és limitat. Tot i l'existència d'aquests blocs, la caixa de control no incorpora aquesta opció de fàbrica sinó que s'ha de recórrer a un producte d'UR+. També s'ha pogut comprovar que els blocs de terminals per connectar-hi els diferents elements: sensor i actuadors, no són del tot còmodes si s'hi han de dur a terme modificacions constants.

Finalment, es valora de manera positiva la incorporació d'aquest robot que difereix dels actuals que hi ha a laboratori. D'aquesta manera es disposa d'un equipament més complet que permet, als estudiants del grau d'enginyeria en electrònica industrial i automàtica, desenvolupar camps d'experimentació més amplis afavorint el seu aprenentatge i possibilitats laborals.

A nivell persona, esmentar que ha estat un treball que m'ha captivat des del principi al suposar un repte donat que no tenia nocions i aquests s'han hagut d'anar aprenent a mesura que sorgien els problemes.



## Pressupost i/o Anàlisi Econòmica

A continuació es mostra l'anàlisi econòmica referent al cost del projecte sense tenir en compte el cost del robot i la taula de muntatge donat que no s'han adquirit per dur a terme el projecte sinó que el projecte s'ha dut a terme a partir de l'adquisició del robot.

En la següent taula es mostren els costos relacionats amb els materials per l'elaboració dels diferents components i dispositius:

**Taula 0.1.** Llista materials i els costos associats.

Nom	Proveïdor	Referència	Unitats	(€/un)	Total (€)
<b>CAIXA SIMULADOR SENSORS DIGITALS</b>					
Interruptor de palanca SPDT, Funcionamiento Enclavamiento	RS comp.	448-0747	3	2.052	6.16
Botón Pulsador Plateado IP65acción momentánea, No, 2 A	RS comp.	690-3255	1	10.3	10.30
<b>ELEMENT TERMINAL</b>					
Imán elevador electroimán 12 vcc de 2,5 kg 5.5lb 20x15m m eléctrica	Amazon	B00CFR1HW6	1	12.97	12.97
Font alimentació 12V 2A	RS comp.	175.3324	1	12.20	12.20
Relé sin enclavamiento Siemens LZX, 4PDT, bobina 24V dc, Enchufable	RS comp.	758-6891	1	10.76	10.76
Zócalo de relé	RS comp	403-285	1	6.34	6.34
3D Warhorse PLA Filament Silver, PLA Filament 1.75mm	Amazon	B07CPT199J	1	17.22	17.22
<b>SUPORT POLYSCOPE</b>					
PERFIL ANGULAR ALUMINI ANODITZAT PLATA 20x20 2m	Fes més	295207002	1	3.75	3.75
METACRILAT TRANSPARENT	Fes més	247332000	1	13.49	13.49
REBLONS D'ALUMINI CONMETALL 50 unitats	Fes més	203415000	1	2.95	2.95
<b>CAIXA CONNEXIONS BANANA</b>					
Connectors bannana femella	RS comp.	886-0917	24	0.635	15.24
Cable 1.5mm (200m) blau	Guerin		1	30.5	30.50
Puntera hueca de crimpado RS PRO, Aislado, Pin de 8mm, Rojo 18AWG 1.5mm²	RS comp.	458-702	150	0.047	7.05

Canalización Ranurada para Cable RS PRO, Gris, PVC, Abierto, Canaleta de panel ranurada, 25 mm 30mm	RS comp.	758-9279	1	6.89	6.89
Carril DIN RS PRO, En U, Plano, long. 500mm, anch. 35mm, alt. 7.5mm	RS comp.	467-406	1	4.73	4.73
Caja de conexiones ABB 1SL0856A00, Termoplástico, Gris, 220mm, 170mm, 80mm, 200 x 170 x 80mm, IP65	RS comp.	455-071	1	10.94	10.94
Conector Recta Montaje Aéreo 8 pines, contactos Hembra	RS comp.	111-5759	2	7.39	14.78
Conector Recta Montaje Aéreo 8 pines, contactos Macho	RS comp.	111-5765	2	8.71	17.42
<b>PANEL DE COMANDAMENT ALTERNATIU</b>					
Interruptor giratorio, 3 Posiciones, tensión máx. 600 V ac/dc, corriente máxima 6 A ac, 550 mA dc Harmony XB5	RS comp.	609-6057	2	31.57	63.14
Interruptor giratorio, 2 Posiciones, tensión máx. 600 V ac/dc, corriente máxima 6 A ac, 550 mA dc Harmony XB5	RS comp.	609-6035	1	21.43	21.43
Pulsador Schneider Electric serie Harmony XB4 Verde, Momentáneo NO	RS comp.	330-8638	1	15.58	15.58
Pulsador Schneider Electric serie Harmony XB4 Negro de 22mm de diámetro, Momentáneo NO	RS comp.	330-8616	1	15.58	15.58
<b>MATERIALS VARIS</b>					
Fusta inferior (DM) 120 X 76 cm	Fes Més	80025000	1	34.60	34.60
Tornillo Allen, Acero Inoxidable, Cabezal con Casquillo Hexagonal, M6 x 20mm 1mm	RS comp.	281-120	1	20.21	20.21
Tuercas cuadradas RS PRO M6 10mm Acero Galvanizado brillante	RS comp.	837-307	1	15.32	15.32
UGREEN Adaptador de Red USB 3.0 a Gigabit Ethernet, USB 3.0 a RJ45 Ethernet	Amazon	B00MYTSN18	1	15.99	15.99
<b>SUBTOTAL</b>					<b>405.54 €</b>

En la taula 0.2 es mostren els costos amortitzables associats al cost dels equips i llicències dels programaris utilitzats:

**Taula 0.2.** Costos amortitzables.

Concepte	Preu (€)
Ordinador	1200.00
Llicència programes informàtics [Solidworks + Microsoft Office + Matlab (versió estudiant)]	7379.00
<b>TOTAL 4 ANYS</b>	<b>8579.00</b>
<b>SUBTOTAL AMORTITZABLE ANUAL</b>	<b>2,144.75 €</b>

A continuació es detallen els costos dels recursos humans que ha comportat l'elaboració d'aquest treball:

**Taula 0.3.** Costos recursos humans.

Concepte	Hores	(€/hora)	Preu aplicat	Total (€)
1. Planificació projecte i organització	15	30	Eng. Jun.	450
2. Formació i aprenentatge autònom previ	16	30	Eng. Jun.	480
3. Disseny layout i estacions robots amb CAD	74	30	Eng. Jun.	2220
4. Disseny components + plànols + esquemes elèctrics	40	30	Eng. Sen.	1200
5. Construcció elements dissenyats	48	30	Eng. Sen.	1440
6. Muntatge i posada en marxa robot	24	30	Eng. Sen.	720
7. Jornades de portes obertes				
7.1 Disseny estació	5	30	Eng. Sen.	150
7.2 Construcció i muntatge de les diferents tasques	32	30	Eng. Sen.	960
7.3 Programació de les tàsquest	16	30	Eng. Sen.	480
7.4 Assistència a les jornades de portes obertes	8	30	Eng. Sen.	240
8. Estudi i programació accés remot	18	30	Eng. Sen.	540
9. Aplicacions i programació comunicació per Socket	56	30	Eng. Sen.	1680
10. Elaboració manuals i vídeos demostratius	32	30	Eng. Sen.	960
11. Memòria				
11.1 Redacció memòria	110	30	Eng. Jun.	3300

11.2 Revisions parcials i control projecte	16	30	Eng. Jun.	480
11.3 Revisions parcials i control projecte	16	60	Eng. Sen.	960
11.4 Revisió final	5	60	Eng. Sen.	300
<b>TOTAL HORES</b>	<b>516</b>		<b>SUBTOTAL</b>	<b>16,560.00 €</b>

Finalment, el cost total del projecte sense tenir en compte el preu del robot que oscil·la als 30,000 € i la taula que té un preu aproximat de 1000 €, és de 19,110.29 €. Els preus de referència s'han extret d'un document d'UPCcommons [41].

**Taula 0.4.** Costos totals del projecte.

Concepte	Preu (€)
Cost components i materials	405.54
Despeses amortitzables	2144.75
Cost recursos humans	16560.00
<b>TOTAL</b>	<b>19,110.29€</b>

## Bibliografia

- [1] Universal Robots, “Universal Robots e-Series Manual de usuario.”
- [2] “Universal Robots Academy.” [Online]. Available: <https://academy.universal-robots.com/es/>. [Accessed: 14-Jun-2020].
- [3] “Universal-Robots and Robotiq distributor.” [Online]. Available: <https://www.zacobria.com/>. [Accessed: 20-Jun-2020].
- [4] “IEEE Xplore.” [Online]. Available: <https://ieeexplore.ieee.org/Xplore/home.jsp>. [Accessed: 20-Jun-2020].
- [5] Viquipèdia, “Robot - Viquipèdia, l'enciclopèdia lliure,” 10-Mar-2020. [Online]. Available: <https://ca.wikipedia.org/wiki/Robot#Etimologia>. [Accessed: 21-Mar-2020].
- [6] S. Tornil Sin, “T1 . Introducción a la robótica industrial,” 2018.
- [7] IEEE, “Types of RobotS.” [Online]. Available: <https://robots.ieee.org/learn/types-of-robots/>. [Accessed: 01-Apr-2020].
- [8] P. S. Inc, “What are the Different Types of Collaborative Robots?,” 02-Oct-2018. [Online]. Available: <https://www.processsolutions.com/what-are-the-different-types-of-collaborative-robots/>. [Accessed: 23-Mar-2020].
- [9] B. Masó Clota and C. Fernández Laborda, “INFORME ACTIVITAT DIRIGIDA: Seguretat industrial,” vol. 1, pp. 2019–2020, 2019.
- [10] C. Bittner, H. Bode, and A. Christ, “The Safety Compendium,” 2013.
- [11] S. Tornil Sin, “T2. El robot manipulador industrial,” pp. 1–64, 2018.
- [12] T. Bajd and M. Mihelj, “INVERSE GEOMETRY AND WORKSPACE OF ROBOT MECHANISMS,” 2010.
- [13] A. Joubair, “What are Accuracy and Repeatability in Industrial Robots?,” 07-Oct-2014. [Online]. Available: <https://blog.robotiq.com/bid/72766/What-are-Accuracy-and-Repeatability-in-Industrial-Robots>. [Accessed: 05-Apr-2020].
- [14] G. Ellis, “Control-System Background Feedback Sensor.”
- [15] Engineer On A Disk, “Industrial Robotics.” [Online]. Available: [http://engineeronadisk.com/V2/book\\_integration/engineeronadisk-14.html](http://engineeronadisk.com/V2/book_integration/engineeronadisk-14.html).
- [16] F. ARRIBAS, “NORMAS TECNICAS EN SEGURIDAD ROBOTICA.”
- [17] P. Europeu, “DIRECTIVA 2006/42/CE DEL PARLAMENTO EUROPEO Y DEL CONSEJO,” May 2006.
- [18] “PL (Performance Level) : Safety Knowledge.” [Online]. Available:

- <https://www.keyence.com/ss/products/safetyknowledge/performance/level/>. [Accessed: 20-Jun-2020].
- [19] “What Is IEC 61508? And Safety Integrity Level (SIL).” [Online]. Available: <https://www.perforce.com/blog/qac/what-iec-61508-how-determine-safety-integrity-level-sil-basics>. [Accessed: 21-Jun-2020].
- [20] OMRON EUROPE B.V., “Guía de Seguridad.”
- [21] E. Dominguez, “Risk Assessment Process for Collaborative Robot Applications.”
- [22] O. Görnemann, “ISO TS15066-ROBOTS COLABORATIVOS MITOS Y REALIDADES,” Dec. 2015.
- [23] Movicontrol, “Robots colaborativos en la industria 4.0 ,” 11-Jul-2018. [Online]. Available: <https://movicontrol.es/robots-colaborativos/>. [Accessed: 07-Apr-2020].
- [24] Universal Robots, “Acerca de Universal Robots | Valores - Equipo - Historia - Organización.” [Online]. Available: <https://www.universal-robots.com/es/acerca-de-universal-robots/>. [Accessed: 31-Mar-2020].
- [25] “La filosofía de Universal Robots es hacer la robótica accesible a todo el mundo - infoPLC.” [Online]. Available: <https://www.infopl.net/plus-plus/entrevistas/item/104986-jacob-pascual-universal-robots>. [Accessed: 08-Jun-2020].
- [26] A. Swales, “Open Modbus / Tcp Specification,” no. March, pp. 1–26, 1999.
- [27] “PROFINET.” [Online]. Available: <https://www.profibus.com/technology/profinet/>. [Accessed: 07-Jun-2020].
- [28] M. Popp and P. Wenzel, “PROFINet - Linking worlds,” *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, vol. 2, pp. 519–520, 2001.
- [29] A. Quesada Arencibia, O. Déniz Suárez, and F. J. Santana Pérez, “Comunicación mediante sockets,” 2006.
- [30] Z. Lars Skovsgaard, “Introduction to Universal-Robots Script Programming | UR Forum-Help-Q&A.” [Online]. Available: <https://www.zacobria.com/universal-robots-knowledge-base-tech-support-forum-hints-tips/universal-robots-script-programming/>. [Accessed: 11-Apr-2020].
- [31] Universal Robots, “The URScript Programming Language,” p. 30, 2013.
- [32] MIR, “MiR100 UR Interface,” pp. 1–17, 2016.
- [33] “What is the Best Programming Language for Robotics?” [Online]. Available: <https://blog.robotiq.com/what-is-the-best-programming-language-for-robotics>. [Accessed: 07-Jun-2020].
- [34] “Tool Center Point Control | FANUC America.” [Online]. Available: <https://www.fanucamerica.com/products/cnc/cnc-technology/tool-center-point-control>. [Accessed: 09-Jun-2020].



- [35] B. Masó Clota and P. Feliu Cuberes, “Les galgues extensomètriques,” pp. 1–16, 2018.
- [36] Viquipèdia, “Protocol de transferència de fitxers (FTP),” 22-Mar-2020. [Online]. Available: [https://ca.wikipedia.org/wiki/Protocol\\_de\\_transferència\\_de\\_fitxers](https://ca.wikipedia.org/wiki/Protocol_de_transferència_de_fitxers). [Accessed: 24-Mar-2020].
- [37] U. Robots, “Dashboard Server e-Series, port 29999 - 42728.” [Online]. Available: <https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/dashboard-server-e-series-port-29999-42728/>. [Accessed: 24-Mar-2020].
- [38] B. Masó, P. Ponsa, and S. Tornil, “Pautas para el diseño de tareas cooperativas persona-robot en ámbito académico.” .
- [39] “IOFirebug Engine for Universal Robots | 4Each.” [Online]. Available: <https://4each.cz/iofirebug-engine-for-universal-robots>. [Accessed: 15-Jun-2020].
- [40] M. Faccio, M. Bottin, and G. Rosati, “Collaborative and traditional robotic assembly: a comparison model,” *Int. J. Adv. Manuf. Technol.*, vol. 102, no. 5–8, pp. 1355–1372, 2019.
- [41] “ANEXO I I. COSTE DEL PROYECTO.” [Online]. Available: [https://upcommons.upc.edu/bitstream/handle/2099.1/17004/Anexo I - COSTE DEL PROYECTO.pdf?sequence=10&isAllowed=y](https://upcommons.upc.edu/bitstream/handle/2099.1/17004/Anexo_I_-_COSTE_DEL_PROYECTO.pdf?sequence=10&isAllowed=y). [Accessed: 20-Jun-2020].



## Annex

A continuació s'adjunten els annex corresponents a:

### A1. Codi Matlab complet capítol 9.2.1

```
%% Inicialització sistema de vídeo i port sèrie (OPCIÓ 2)
% Inici sistema de vídeo amb captura d'imatges en temps real. Els drivers poden
variar segons el model i resolució de la càmera.
% delete(imagfind)
% vidobj = videoinput('winvideo', 1, 'YUY2_320x240');
% set(vidobj, 'ReturnedColorSpace', 'grayscale');
% triggerconfig(vidobj, 'manual');
% colormap(gray);
% start(vidobj);

%% ----- Comunicació per Socket amb robot UR3e -----

% Configurar IP robot i ordinador.
% Posar la IP de l'ordinador en el programa del robot.
% Posar la IP del robot en el programa del Matlab.
% Primer executar el programa del Matlab
% -----

%% Inici del codi
clear all % Borrar variables
clc

% Connexio amb el host
IP_robot = '192.168.199.128';
Port = 50; % Port de connexió.
s = tcpip(IP_robot, Port, 'NetworkRole', 'server'); % Definim com a servidor.
disp('Executa el programa del robot');
fopen(s); % Obrir la comunicació.
disp('Connexió establerta!');

%% Bucle principal del programa (s'executa de forma indefinida)
fin = 0;
while (fin==0)
    j=str2double(fscanf(s, '%c')); % S'espera la recepció del sensor inductiu 1
    per iniciar la seqüència de processat (s'espera rebre un 1.0 i es passa a 1)
    if j == 1

        %% (OPCIÓ 1)
        %% Lectura de la imatge des de fitxer extern.
        % fprintf('\n');
        str = input('Escriu el nom i extensió del fitxer: ', 's');
        % Es sol·licitarà posar el nom i extensió del document d'imatge.
        I = imread(str);
        fprintf(s, '(9.0)'); % Enviament senyal activació cinta
        j=0;

        %% (OPCIÓ 2)
        %% Captura imatge amb Webcam
        % Captura d'una imatge (foto)
        %I = getsnapshot(vidobj);
        % Visualització
        %figure, imshow(I);
```

```

% Desconnexió entrada vídeo
%stop(vidobj);
%delete(vidobj);

%% CODI DETERMINACIÓ DE LA PEÇA
% Retallem la imatge per millorar el procés d'anàlisis.
imatge = imcrop(I,[30 75 225 190]);

% Apliquem un petit ajust a les imatges:
K = imadjust(imatge,[0.55 0.9],[]);

% Escollir automàticament el llindar amb el mètode Otsu
level = graythresh(K);
imatgebn = im2bw(K, level);

%% Aplicació Morfologies per obtenir una imatge amb els objectes
definites i separats

% Omplir a 1 els forats de les regions de la imatge
Ifilled = imfill(imatgebn, 'holes');

% Definició de la figura a aplicar
sesoft = strel('disk',10); % suau
sehard = strel('disk',25); % fort

% Apertura morfològica
Iopenned = imopen(Ifilled, sesoft);

% Erosió forta de la imatge
IM1 = imerode(Iopenned,sehard);

% Erosió suau de la imatge
IM2 = imerode(IM1,sesoft);

%% ETIQUETATGE
[L,n]=bwlabel(IM1);

%% Obtenció de la imatge neta
reg = regionprops(IM1,'Area');

% Després d'analitzar totes les àrees, eliminem totes les que no estan
% incloses en l'interval escollit.
BW2 = bwareafilt(IM1,[1200 5000]);
%figure, imshow(BW2,[]);

% Variables per la representació d'etiquetes
imshow(label2rgb(BW2));
prop= regionprops(BW2);

%% PROPIETATS A ESTUDIAR DE LES IMATGES

% Propietats regionprops:
A=regionprops(BW2,'Area');
P=regionprops(BW2,'Perimeter');

% Al tenir només un objecte en al imatge, obtenim els seus valors de
cada
% propietat.
Area=A(1).Area(1);
Perimetre=P(1).Perimeter(1);
Circularitat=(4*pi*Area)/Perimetre^2;

%% PROGRAMA DE DETECCIÓ DE PECES A PARTIR DE LES PROPIETATS ESTUDIADAES

```

```

% Si compleixen la condició de circularitat entrem en la condició
if Circularitat>0.5

    % Detecció peces quadrades
    if Area>1250 && Area<2300 && Perimetre<180
        tipus_pecas=1;

        % Detecció peces rectangulars petites
        elseif Area>2200 && Area<3100 && Circularitat>0.68 &&
Circularitat<0.78
            tipus_pecas=2;

        % Detecció peces rectangulars grans
        elseif Perimetre>200 && Circularitat>0.5 && Circularitat<0.7
            tipus_pecas=3;

        % Detecció peces circulars
        elseif Circularitat>0.85 && Area>3000
            tipus_pecas=4;

        % Exclosió de peces no corresponents a cap de les anteriors
    else
        tipus_pecas=0;
    end
else
    tipus_pecas=0;
end

%% ENVIAR SENYAL AL ROBOT UNA VEGADA DETERMINADA LA PEÇA
switch (tipus_pecas)
    case 1
        fprintf(s,'(1.0)'); % Tasca 1 Seleccionada
        disp('Peça detectada: quadrada');
    case 2
        fprintf(s,'(2.0)') % Tasca 2 Seleccionada
        disp('Peça detectada: rectangular petita');
    case 3
        fprintf(s,'(3.0)') % Tasca 3 Seleccionada
        disp('Peça detectada: rectangular gran');
    case 4
        fprintf(s,'(4.0)') % Tasca 4 Seleccionada
        disp('Peça detectada: circular');
    otherwise
        fprintf(s,'(5.0)') % Tasca 5 Seleccionada
        disp('Peça detectada: defectuosa, descartar peça');
end

%% Visualització diferents resultats:
subplot(2,3,1)
imshow(I)
title('Imatge Original')
subplot(2,3,2)
imshow(imatge)
title('Imatge Retallada')
subplot(2,3,3)
imshow(Ifilled)
title('Omplim els forats')
subplot(2,3,4)
imshow(IM1)
title('Erosió forta')
subplot(2,3,5)
imshow(BW2)

```

```

title('Filtrat de mida')
subplot(2,3,6)
imshow(label2rgb(BW2))
title('Etiquetes colorejades')
subplot(2,3,6)
hold on
for n=1:length(prop)
    rectangle('Position',prop(n).BoundingBox,'EdgeColor','g',
        'LineWidth',2)
    x=prop(n).Centroid(1);
    y=prop(n).Centroid(2);
    plot(x,y,'*')
    % text(x,y,'O')
end
hold off
title('Localització etiquetes')

end
end

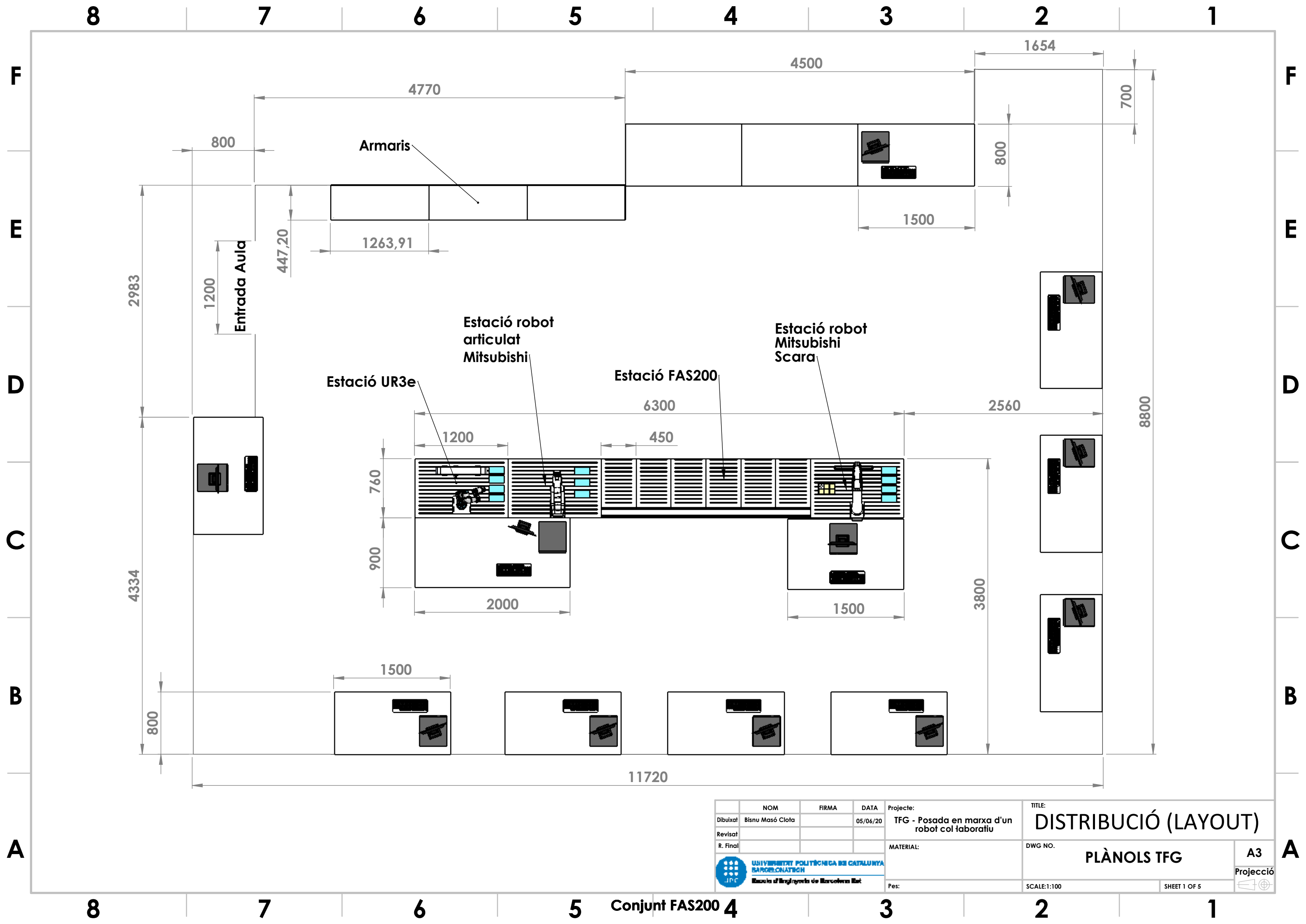
%% Tancar el port sèrie (finalitzar comunicació PC - Controlador)
% fclose(s);
% delete(s);
% clear s;

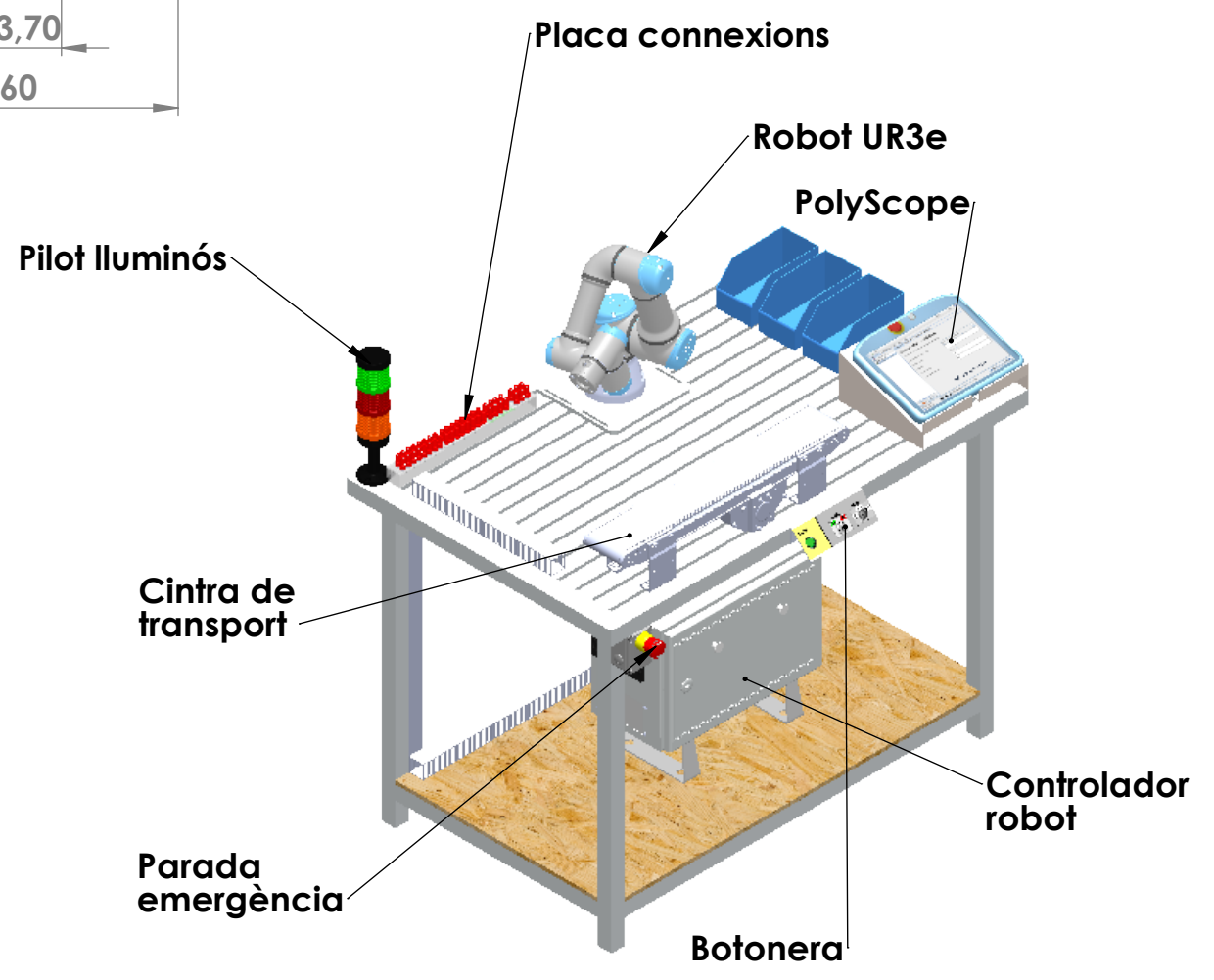
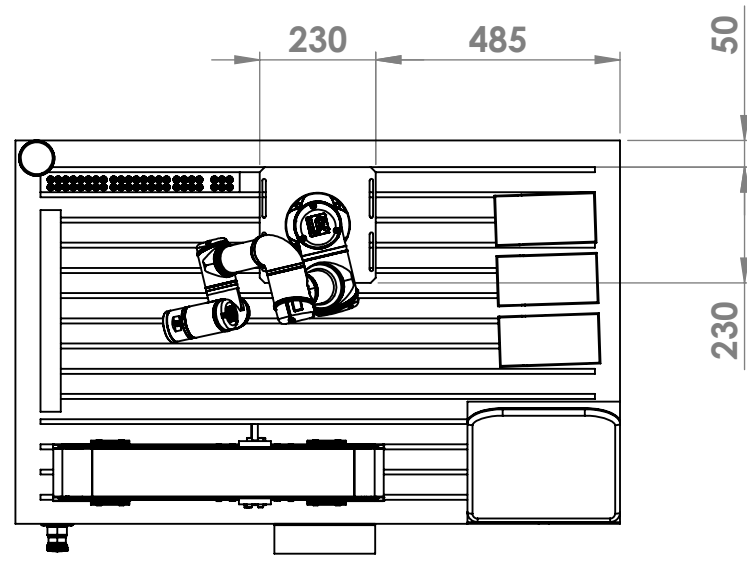
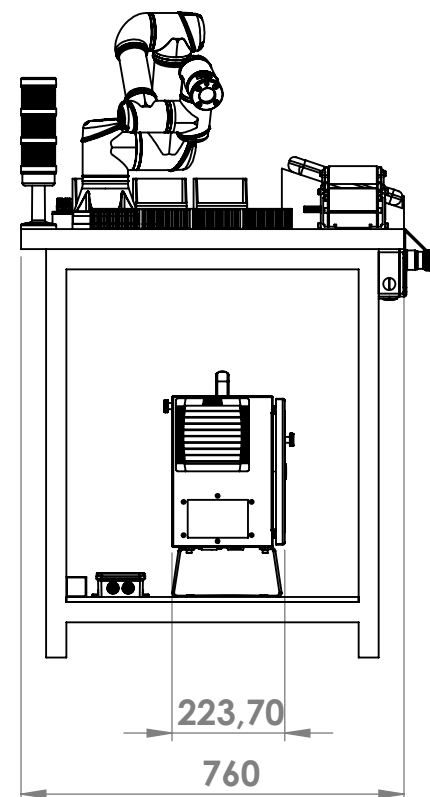
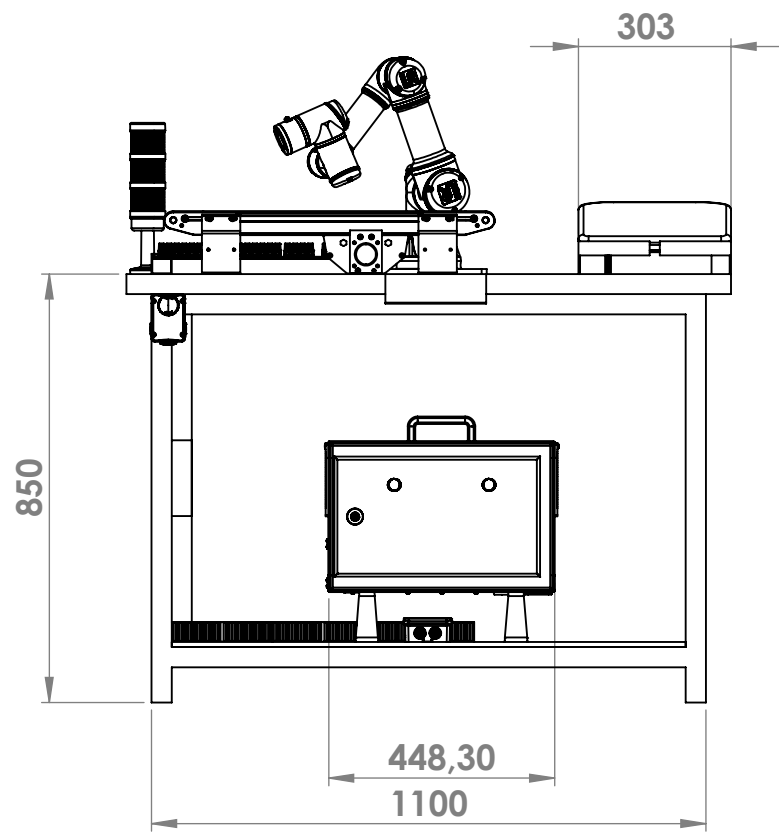
```



## A2. Plànols

## A3. Esquemes elèctrics

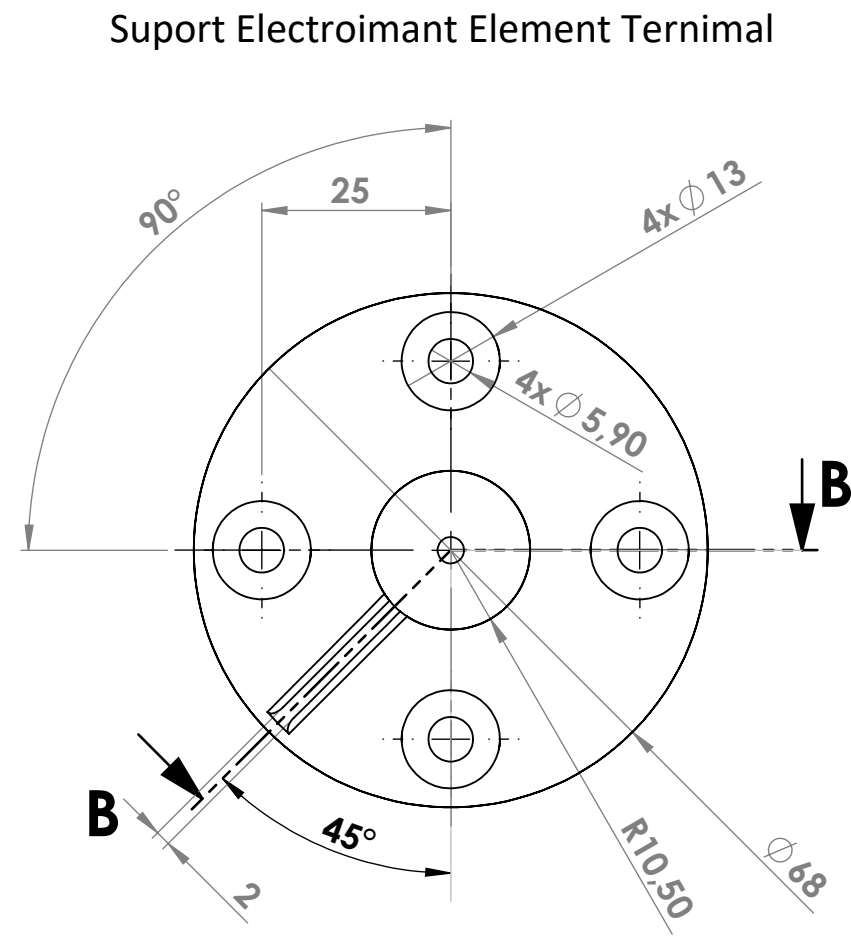
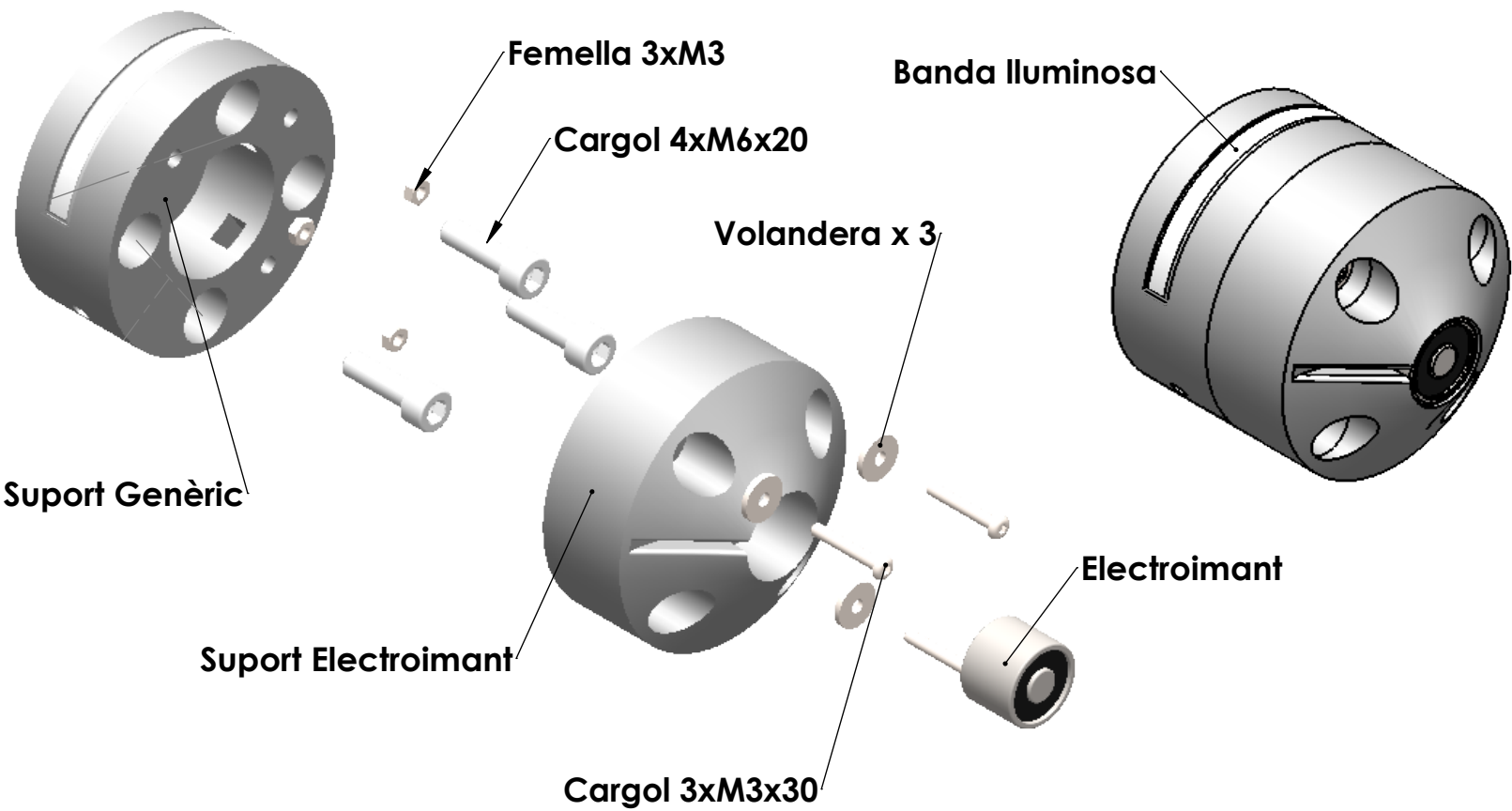
## A3. Datasheet Robot Ur3e



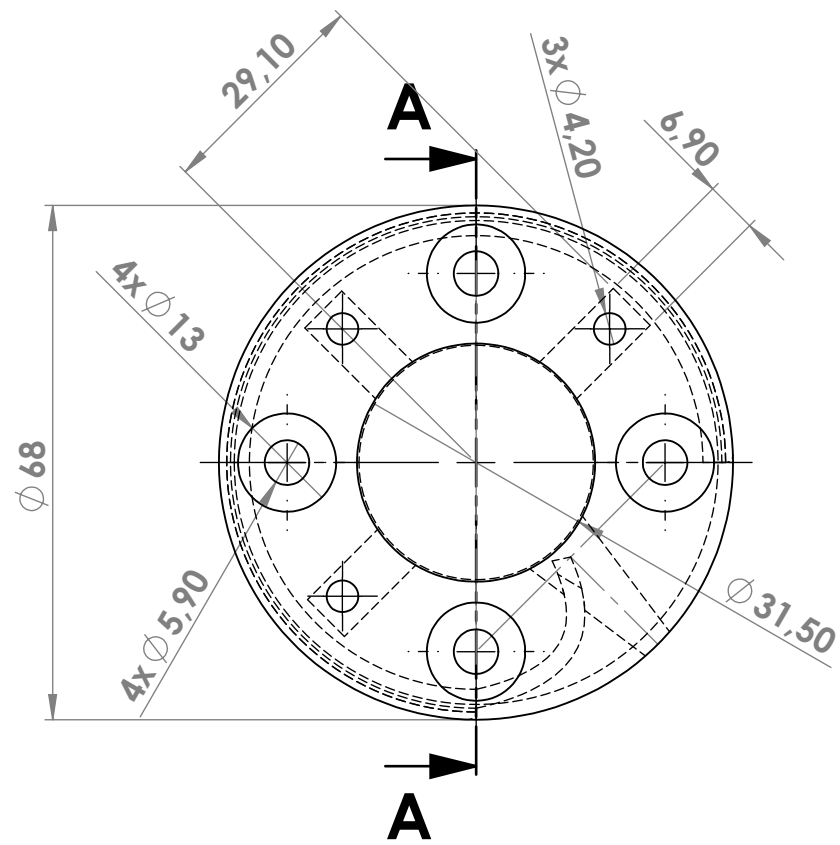


NOM		FIRMA		DATA		Projecte:		TITLE:	
Dibuixat	Bisnu Masó Clota				05/06/20	TFG - Posada en marxa d'un robot col·laboratiu		AESTACIÓ ROBOT UR3e	
Revisat									
R. Final									
 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATÈCNIC Escola d'Enginyeria de Barcelona Est						MATERIAL:		DWG NO.	
						Pes:		SCALE:1:15	
								SHEET 2 OF 5	
								A3 Projecció 	

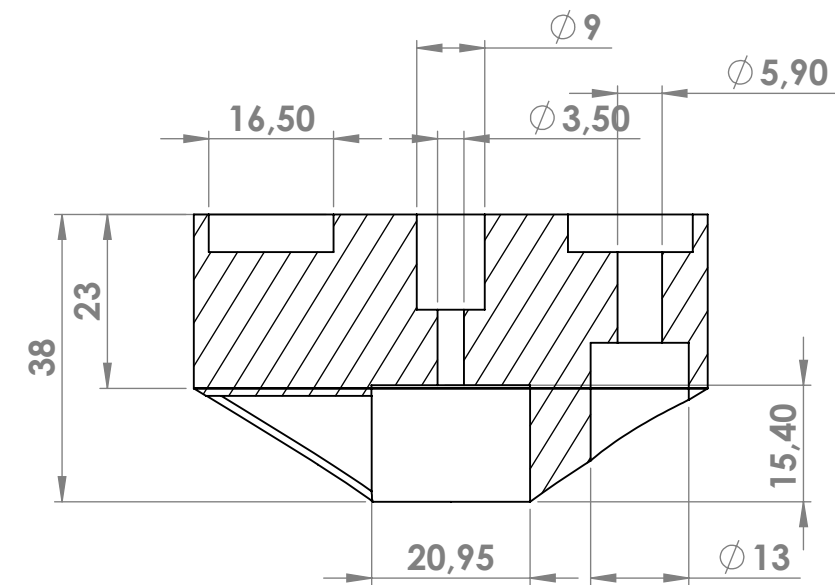
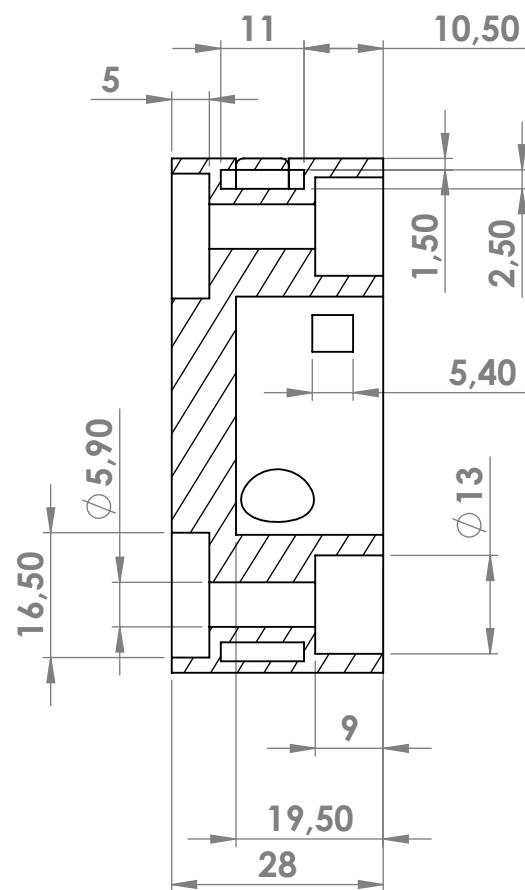




Suport Genèric Element Terminal



**Secció A-A**  
ESCALA 1 : 1



**Secció B-B**  
ESCALA 1 : 1

NOM		FIRMA	DATA	Projecte:	TITL:	
Dibuixat	Bisnu Masó Clota		05/06/20	TFG - Posada en marxa d'un robot col·laboratiu	ELEMENT TERMINAL	
Revisat				MATERIAL:	DWG NO.	A3
R. Final				PLA (Àcid polilàctic)	PLÀNOLS TFG	Projecció
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATÈCH		Escuela d'Enginyeria de Barcelona Est		Pes:	SCALE:1:1	SHEET 3 OF 5



F

E

D

C

B

A

F

E

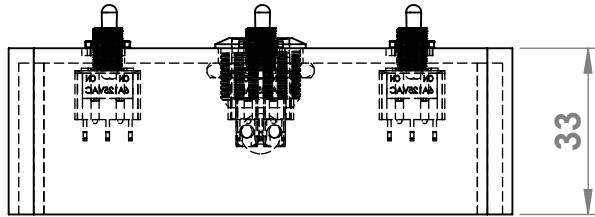
D

C

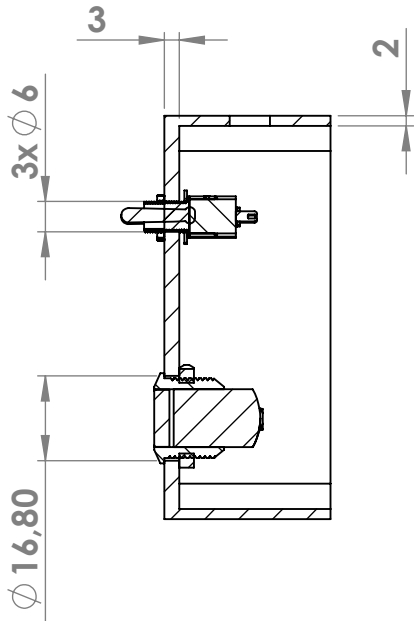
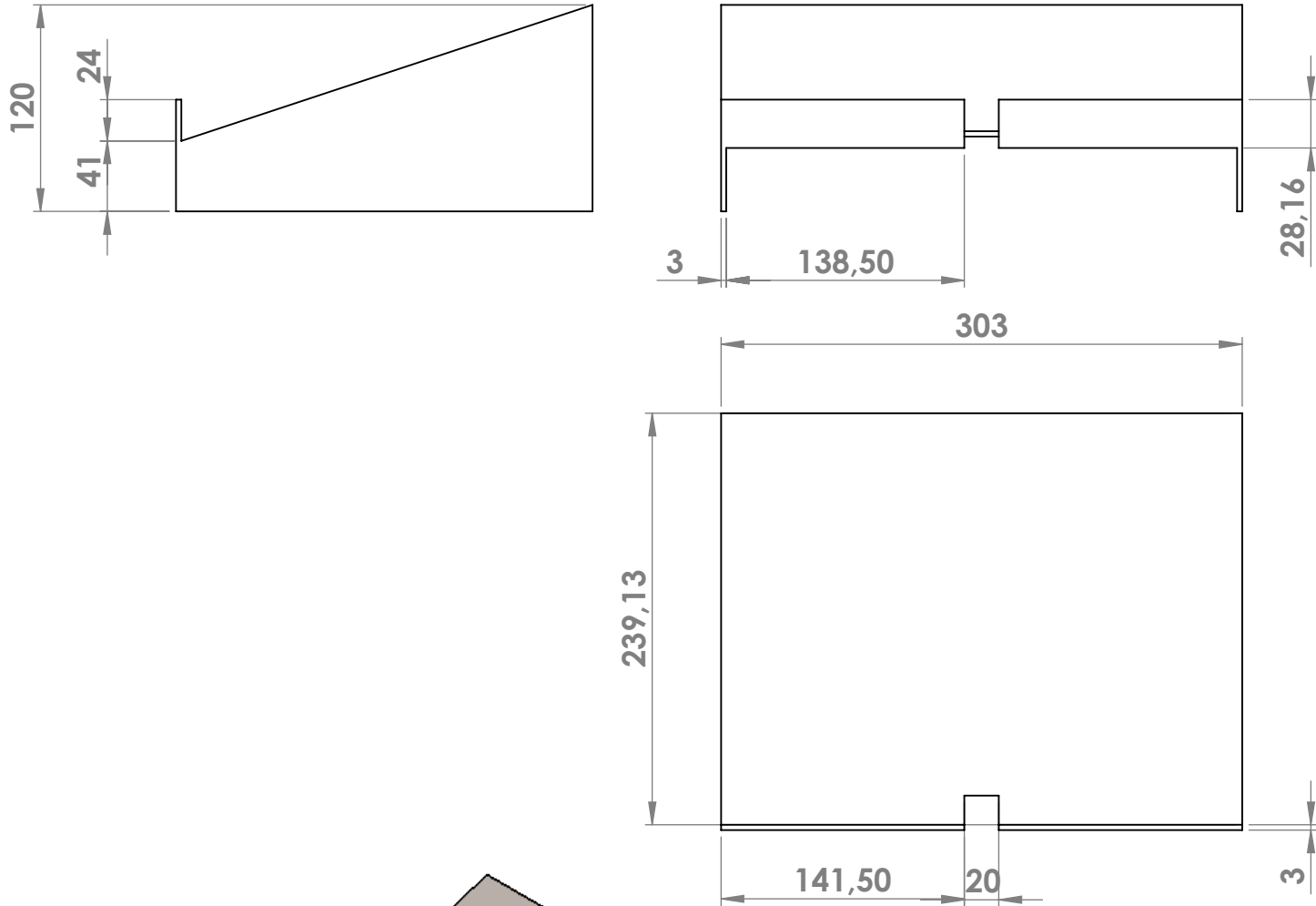
B

A

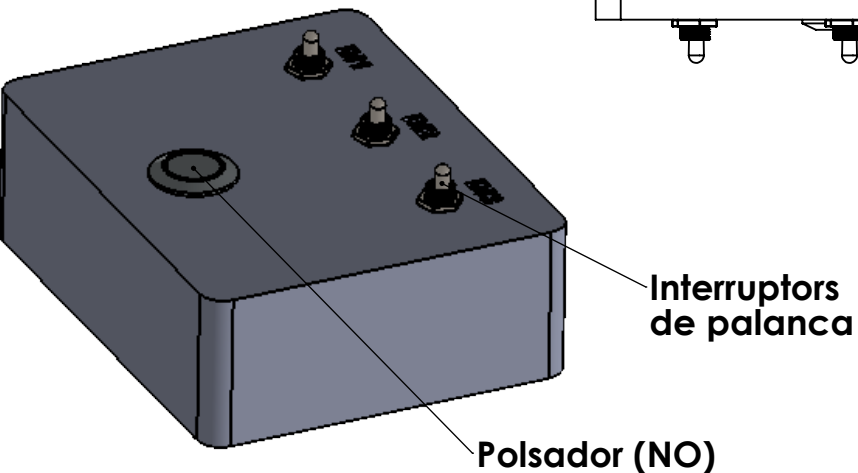
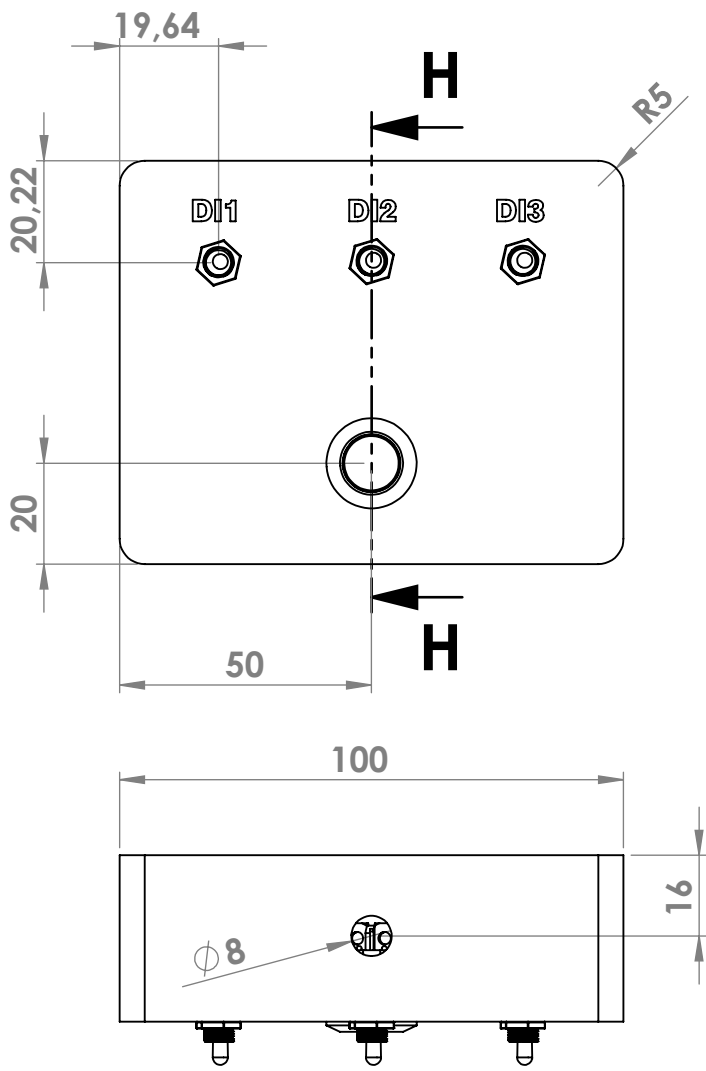
Simulador sensors digitals  
(Escala 1:1.5)



Suport consola de programació  
(Escala 1:4)



Secció H-H  
ESCALA 1 : 1.5



Interrupctors de palanca

Polsador (NO)

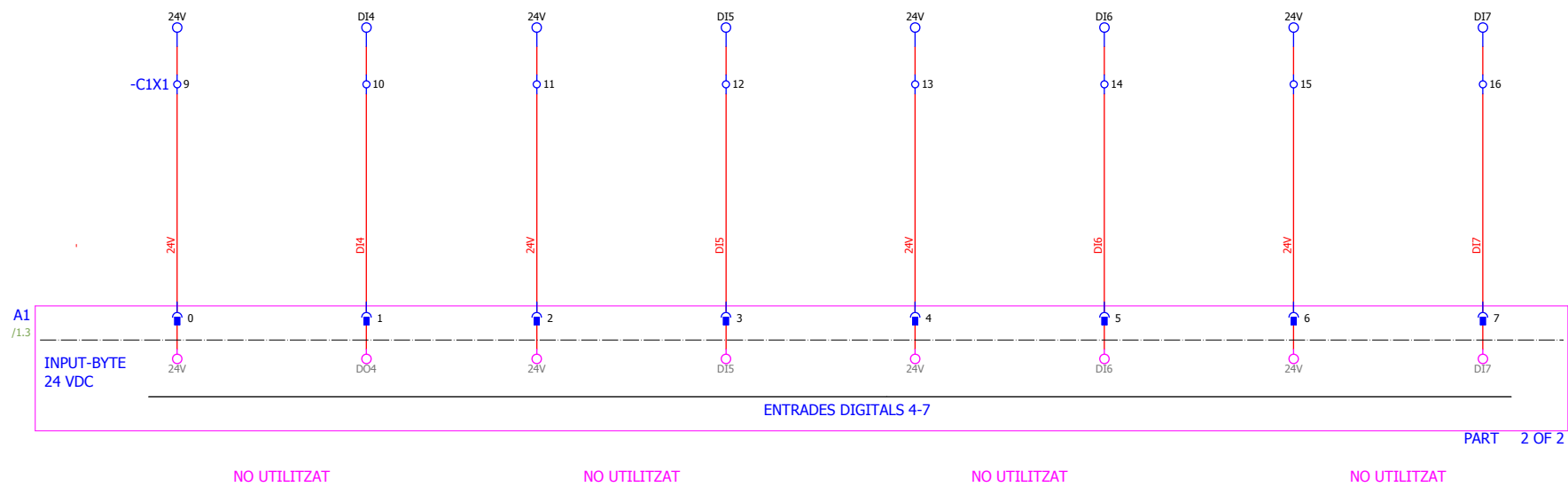
	NOM	FIRMA	DATA	Projecte:	TITLE:
Dibuixat	Bisnu Masó Clota		05/06/20	TFG - Posada en marxa d'un robot col·laboratiu	ACCESSORIS 2
Revisat					
R. Final					
				MATERIAL:	DWG NO.
				PLA (Àcid polilàctic) Metacrilat	PLÀNOLS TFG
				Pes:	SCALE:1:5
					SHEET 5 OF 5



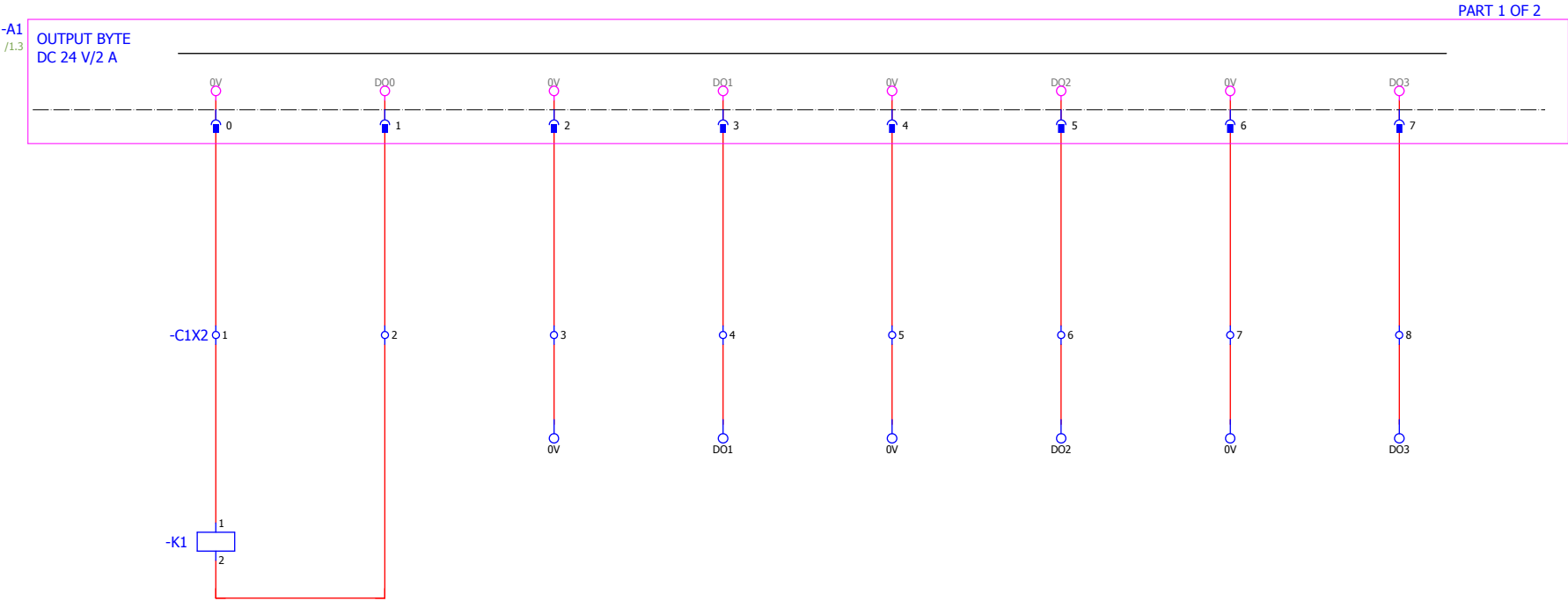








			Fecha	17/05/2020	EPLAN		EPLAN Software & Service GmbH & Co. KG	ENTRADES DIGITALS (4 a 7)			=				
			Resp.	USUARI									+		
			Probado		Esquemes TFG UR3e										
Cambio	Fecha	Nombre	Original		Sustitución por	Sustituido por					IEC_tpl001		Hoja 4		
													Hoja 10		



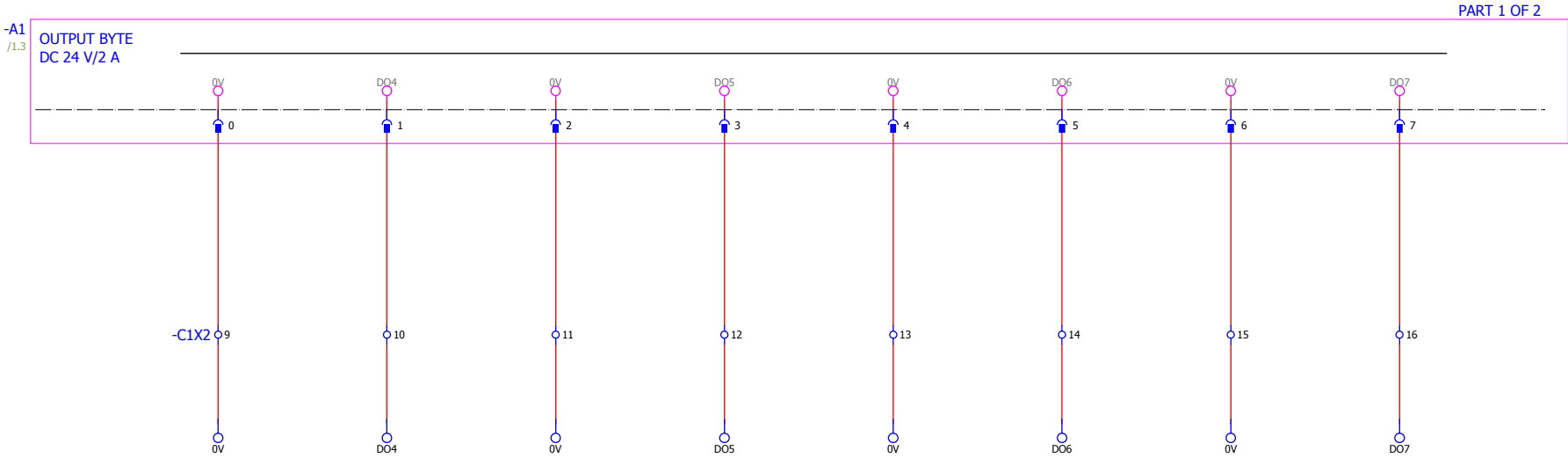
RELÉ ELEMENT TERMINAL

NO UTILITZAT

NO UTILITZAT

NO UTILITZAT





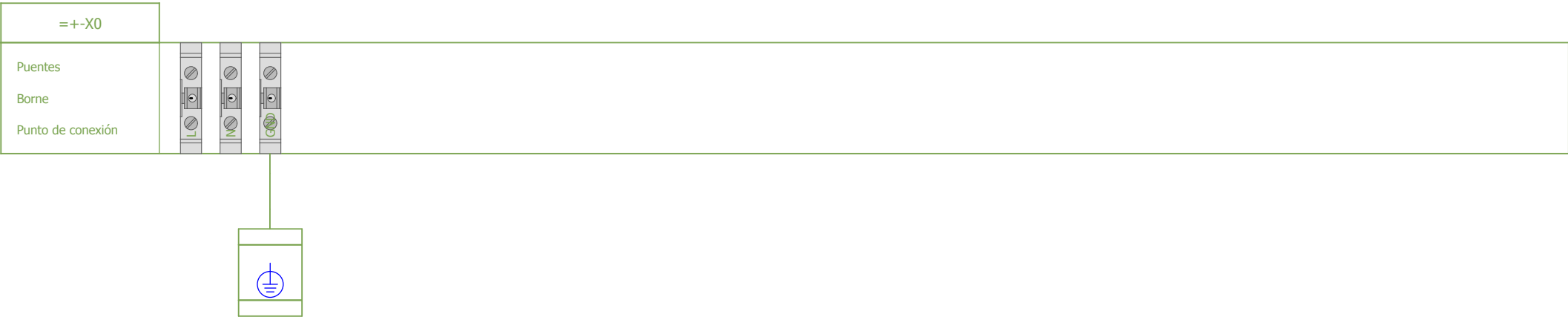






## Plano de conexiones de bornes

## Destinos internos



## Destinos externos

[illegible]

# UR3e

## Ficha técnica

### Rendimiento

Consumo de energía	Aprox. 100W para un programa típico		
Operación de colaboración	17 funciones de seguridad ajustables avanzadas, incluyendo supervisión del código Control remoto de acuerdo con ISO 10218		
Certificaciones	EN ISO 13849-1, Cat.3, PL d, y EN ISO 10218-1		
Sensor F/P - Fuerza, X-Y-Z	Sensor F/P - Par, X-Y-Z		
Rango	30 N	Rango	10 Nm
Resolución	1,0 N	Resolución	0,02 Nm
Precisión	3,5 N	Precisión	0,10 Nm
Rango de temperatura ambiente	0–50°C*		
Humedad	90%RH (sin condensación)		

### Especificación

Carga útil	3kg / 6,6lbs		
Alcance	500mm / 19,7in		
Grados de libertad	6 articulaciones giratorias		
Programación	Interfaz gráfica del usuario PolyScope con pantalla táctil de 12" con soporte		

### Movimiento

Repetibilidad de posición	+/-0,03mm con carga, según ISO 9283	
Movim. del eje del brazo robot.	Radio de acción	Velocidad máxima
Base	± 360°	± 180°/s
Hombro	± 360°	± 180°/s
Codo	± 360°	± 180°/s
Muñeca 1	± 360°	± 360°/s
Muñeca 2	± 360°	± 360°/s
Muñeca 3	Infinita	± 360°/s
Velocidad típica de TCP	1 m/s / 39,4 in/s	

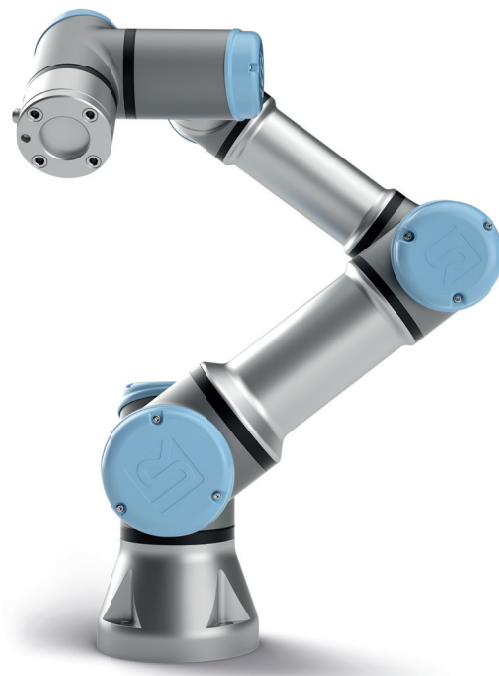
### Funciones

Clasificación IP	IP54		
Clase ISO Sala limpia	5		
Ruido	Menos de 60dB(A)		
Montaje del robot	Cualquier orientación		
Puertos de E/S en herramienta	Entrada digital	2	
	Salida digital	2	
	Entrada analógica	2	
	Salida analógica	0	
	Interface UART (9.6k-5Mbps)		
E/S de fuente de alimentación en herramienta	12V/24V 600mA continuos, 2A por periodos cortos		

### Características físicas

Huella	Ø 128 mm		
Materiales	Aluminio, Plásticos de PP, Acero		
Tipo de conector para herramienta del robot	M8   M8 8-pin		
Long. cable del brazo robótico	6m / 236in		
Peso incluyendo cable	11,2kg / 24,7lbs		

\*El robot puede trabajar dentro del intervalo de temperaturas 0-50 °C. A alta velocidad continua de las articulaciones, la temperatura ambiente se reduce.



## Caja de control

### Funciones

Clasificación IP	IP44	
Clase ISO Sala limpia	6	
Rango de temperatura ambiente	0–50°C	
Puertos de E/S	Entrada digital	16
	Salida digital	16
	Entrada analógica	2
	Salida analógica	2
	Control a 500 Hz, 4 entradas digitales en cuadratura de alta velocidad dedicadas	
E/S de fuente de alimentación	24 V 2A	
Comunicación	Frecuencia de control: 500 Hz ModbusTCP. Frecuencia de señal de 500 Hz ProfiNet y EthernetIP. Frecuencia de señal de 500 Hz Puertos USB: 1 USB 2.0, 1 USB 3.0	
Fuente de alimentación	100-240VAC, 47-440Hz	
Humedad	90%RH (sin condensación)	

### Características físicas

Tamaño de la caja de control (anch. x alt. x prof.)	475mm x 423mm x 268mm / 18,7in x 16,7in x 10,6in
Peso	Máx. 13kg / 28,7lbs
Materiales	Acero

## Consola de programación

### Funciones

Clasificación IP	IP54
Humedad	90%RH (sin condensación)
Resolución de pantalla	1280 x 800 píxeles

### Características físicas

Materiales	Plástico
Peso incluyendo 1m de cable de TP	1,6kg / 3,5lbs
Longitud del cable	4,5m / 17,7in



**UNIVERSAL ROBOTS**

universal-robots.com